

BTS SIO - B1 - Bases de la programmation

ÉVALUATION SEMESTRE 1 - 12 décembre 2024

Ni documents ni matériel autorisés. Durée : 1h45. Coefficient : 2. **Ne répondez que sur votre copie.**

Partie A

A.1 Donnez la sortie du programme du document 1 donné en annexe. On rappelle que la **sortie** d'un programme consiste en ce qui est affiché sur l'écran et rien d'autre. Si vous souhaitez fournir des explications pour clarifier votre pensée (non demandé), elles doivent être fournies à part : **la sortie doit être encadrée et complètement isolée sur votre copie.**

Partie B

Vous intervenez dans la phase de développement initiale d'une application web de type *e-commerce*. Vous développez une version prototype du code qui va permettre la pagination lors de l'affichage d'un catalogue de produits. Ce prototype va être développé dans un premier temps en mode texte sur un terminal. L'ébauche actuelle est donnée dans le document 2 en annexe.

Le programme à compléter devra :

- calculer le nombre de pages total nécessaires à l'affichage de tous les produits ;
- afficher ce nombre de pages total ;
- puis procéder à l'affichage de la totalité des pages une par une.

Le document 3 fourni un extrait de sortie possible.

On commence par travailler sur la méthode qui affiche le contenu d'**une** page. Étant donnés un **numéro de page** et un **nombre de produits par page** (fournis par l'appelant), cette méthode affiche tous les produits correspondants (la partie entre crochets sur l'exemple fourni). Elle récupère les données des produits via la base de données, à laquelle elle a un accès direct dont vous ne devez pas vous soucier.

B.1 Donnez l'**entête** de cette méthode. On rappelle que l'entête est la première ligne de la méthode, commençant par **public static**. **Le corps de la méthode elle-même n'est pas demandé.**

On travaille maintenant sur la méthode **main**. Pour les tests, on fera les hypothèses suivantes :

- il y a 1000 produits dans le catalogue ;
- il faut afficher 25 produits par page ;
- ces valeurs de test devront pouvoir être facilement modifiées par la suite, lorsque les données proviendront d'une base de données ou d'une entrée utilisateur.
- la méthode de la question B.1 a été implémentée et testée.

B.2 Complétez la partie « *Initialisation des variables* » du programme fourni dans le document 2.

B.3 Complétez la partie « *Calcul du nombre de pages total* ».

Le nombre de produits par page sera sélectionné par l'utilisateur grâce à un menu déroulant. Pour pouvoir simuler ce comportement dans notre prototype, on va écrire une méthode `recupérerNbProduitsParPage` permettant de renvoyer un nombre entré au clavier par l'utilisateur. La méthode doit afficher un message explicatif avant de récupérer le nombre, qui doit être compris entre 10 et 100 inclus. On vérifiera que le nombre entré répond à ces critères et on redemandera autant de fois que nécessaire.

B.4 Écrire la méthode `recupérerNbProduitsParPage` (entête + corps).

B.5 Indiquez les modifications à effectuer dans `main` pour utiliser cette méthode plutôt que le nombre `25` fixé auparavant « en dur ».

B.6 Complétez la partie « *Affichage de toutes les pages selon le modèle spécifié* ». Le format du modèle est donné dans le document 3. On rappelle également que la méthode de la question B.1, que vous pouvez utiliser, ne s'occupe pas de l'affichage des entêtes de pages (« PAGE 1 : », etc.).

Partie C

On considère une version du Jeu de l'Oie dans laquelle on a cinq lancers de dé pour la partie (document 4). On souhaite mettre en place les modifications suivantes :

- Lorsqu'on dépasse la case gagnante (ici 20), on recule d'autant de cases que le lancer de dé dépasse ; par exemple, si on est à la case 18 et qu'on fait un 5, on se retrouve non pas à la case 23 mais à la case 17 (on va jusque 20 et on recule).
- Le jeu doit continuer indéfiniment tant que le pion n'arrive pas pile sur la case 20 (par exemple, en étant sur la case 17, il nous faut absolument un 3 pour gagner) ; **il n'y a donc plus de partie perdue.**
- À la fin, on doit afficher le nombre de lancers qui ont été nécessaires pour gagner.

Les trois questions suivantes ne nécessitent pas la présentation du code précis, seulement une explication en une ou deux phrases.

C.1 On ne connaît donc plus le nombre de lancers qu'on va devoir faire pour gagner. À quel genre de modification peut-on immédiatement penser lors de l'analyse de ce nouveau besoin ?

C.2 Quel genre de structure de contrôle Java utilisera-t-on pour traiter le cas spécial « on dépasse la case gagnante » ?

C.3 Expliquez comment vous pouvez traiter la partie « connaître le nombre de lancers qui ont été nécessaires pour gagner ».

Avant d'implémenter ces modifications, on souhaite modulariser le programme existant (doc. 4), c'est-à-dire le découper en méthodes.

C.4 Identifiez deux parties que vous pourriez modulariser et donnez un nom pour chacune des méthodes correspondantes.

C.5 Donnez le code complet des deux méthodes (entêtes + corps).

C.6 Réécrivez la méthode `main` en utilisant les deux méthodes que vous avez définies.

Annexes

Document 1 (partie A)

```
public class QuelleEstLaSortie {
    public static void mod(int celeri) {
        celeri = celeri + 1;
        System.out.println("celerici = " + celeri);
    }

    public static void main(String[] args) {
        double a = 10.5;
        int b = 20;
        String message = "Le résultat est : ";

        boolean result = foo(a, b);
        double navets = bar(b, a);

        int celeri = 10;
        System.out.println("celeri = " + celeri);
        mod(celeri);
        System.out.println("celeri = " + celeri);

        if (result) {
            message = message + "Les sanglots longs des violons";
        } else if (a == b) {
            message = message + "Il a deux trous rouges au côté droit";
        } else {
            message = message + "Sois sage, Ô ma Douleur";
        }

        for (int i = 0; i < 5; i++) {
            System.out.print("Carcassone ");
        }

        System.out.println();
        int c = 3;
        while (c > 0) {
            System.out.println("c = " + c);
            c = c - 1;
        }

        System.out.println(message);
        System.out.println("Les carottes sont cuites : " + navets);
    }

    public static boolean foo(int n1, double n2) {
        boolean b = n1 > n2;
    }
}
```

```

    return b;
}

public static void boz() {
    System.out.println("BOZ !");
}

public static double bar(int a, double b) {
    double result = a - b;
    System.out.println("Combien ? " + result);
    boz();
    return result;
}
}

```

Document 2 (partie B)

```

public static void main(String[] args) {
    // -----
    // Initialisation des variables
    // À COMPLÉTER (question B.2)

    // -----
    // Calcul du nombre de pages total
    // À COMPLÉTER (question B.3)

    // -----
    // Affichage du nombre de pages total - Ne doit pas être modifié
    System.out.println("Nombre de pages total: " + nbPages);
    System.out.println();

    // -----
    // Affichage de toutes les pages selon le modèle spécifié
    // À COMPLÉTER (question B.6)

    // -----
}

```

Document 3 (partie B)

Extrait de sortie possible pour un catalogue de quelques centaines de produits avec un affichage de 25 produits par page :

```

Nombre de pages total: 17

```

PAGE 1 :
[liste des produits 1 à 25]

PAGE 2 :
[liste des produits 26 à 50]

PAGE 3 :
[liste des produits 51 à 75]

...et ainsi de suite...

PAGE 17 :
[liste des produits restants]

Document 4 (partie C)

```
Random generateur = new Random();
int caseGagnante = 20;
int maCase = 0;

for (int i = 1; i <= 5; i++) {
    int lancer = generateur.nextInt(6) + 1;
    System.out.println("Vous avez fait " + lancer);
    maCase = maCase + lancer;
    System.out.println("Vous êtes à la case " + maCase);
}

if (maCase == caseGagnante) {
    System.out.println("Gagné !");
} else {
    System.out.println("Perdu !");
}
```

Document 5 (documentation 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 des entrées utilisateur au clavier (ici de type entier et String)
Scanner sc = new Scanner(System.in);
int n = sc.nextInt();
String s = sc.next();
// Générer un nombre aléatoire entre 0 et 99 inclus
Random rand = new Random();
int aleatoire = rand.nextInt(100);
```