

Examen pratique d'août 2024

SAUT EN HAUTEUR

Le saut en hauteur est une épreuve d'athlétisme qui consiste à franchir une barre horizontale posée entre deux supports verticaux (voir la **figure 1**) en prenant appui sur une jambe. L'athlète qui franchit la barre la plus haute remporte le concours.



Fig. 1 – L'athlète belge Claire Orcel qui réussit un saut à 1m92.

Le concours est régi par deux règles générales : la barre ne peut jamais être descendue, et un athlète est éliminé après trois sauts échoués successifs.

Cependant, lorsqu'un athlète rate son premier essai (ou ses deux premiers essais) à une hauteur déterminée, il a le droit de conserver ses deux essais (ou son dernier essai) pour tenter directement une hauteur supérieure (c'est une *impasse*). Ce n'est qu'au troisième échec successif qu'il est éliminé, ces échecs pouvant donc avoir eu lieu à des hauteurs différentes.

DESCRIPTION DE L'APPLICATION

Référez-vous aux exemples d'exécution donnés en fin d'énoncé afin de mieux comprendre le résultat attendu.

L'application doit permettre, lors d'une épreuve de saut en hauteur, de récolter les données d'un seul athlète afin de proposer à tout moment une vue d'ensemble des sauts effectués par cet athlète. Plus précisément, la vue doit répertorier les différentes hauteurs sautées et les réussites/échecs pour chacune d'elles (voir l'exemple de la **figure 2**).

2m29	2m31	2m33	2m34	2m37	2m38
✗ ✗ ●	●	✗ ✗	●	✗	✗ ✗

Fig. 2 – Un tableau récapitulant les sauts effectués par un athlète.

Dans l'exemple précédent, l'athlète rate ses deux premiers sauts à **2m29** avant de réussir au 3^e essai. Plus tard dans l'épreuve, l'athlète décide de faire l'impasse à **2m33** pour tenter un saut à **2m34** avec son dernier essai. A **2m37**, l'athlète décide également de faire l'impasse pour tenter sa chance à **2m38** avec ses deux derniers essais.

*Retenez bien l'exemple de la **figure 2**, car ce dernier sera réutilisé à plusieurs reprises dans cet énoncé.*

Lorsque l'athlète rate trois sauts successifs, comme c'est le cas ici à **2m37** et à **2m38**, l'application doit indiquer que l'épreuve est terminée pour lui, puis s'arrêter.

AVANT-PROPOS

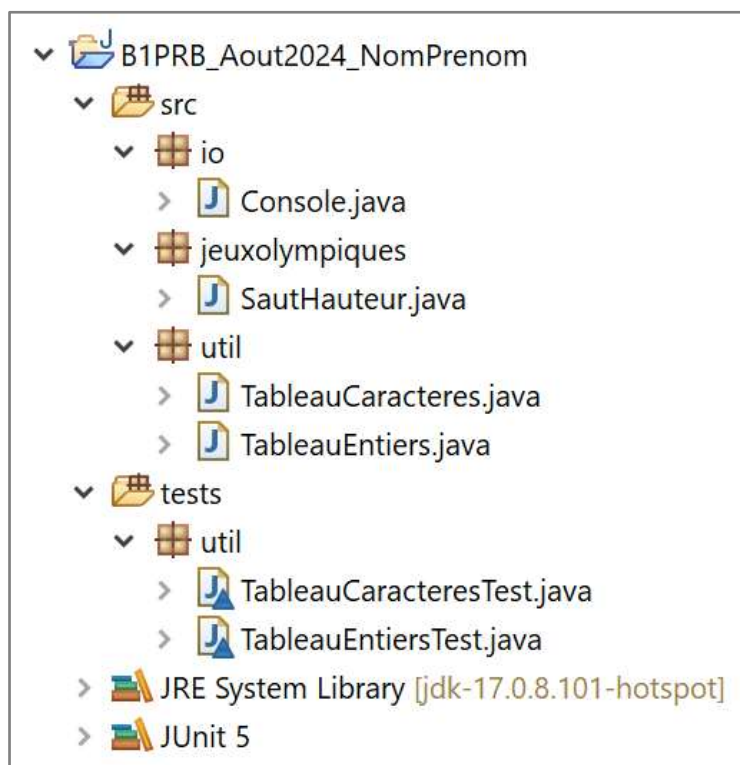
À deux exceptions près, les types **String** et **PrintStream**, vous ne pouvez pas utiliser le paradigme orienté objet pour coder cette application. N'oubliez pas de qualifier toutes vos fonctions de **static** !

PRÉPARATION

1. Dans Eclipse, créez un projet Java nommé **B1PRB_Aout2024_NomPrenom**, où **NomPrenom** doit être remplacé par vos nom et prénom.

N'oubliez pas de vérifier que l'encodage utilisé pour les fichiers est l'**UTF-8**. Pour ce faire, il faut suivre le chemin : *Window > Preferences > General > Workspace > Text file encoding*.

2. Reproduisez exactement l'organisation suivante :



La classe **Console** est à télécharger au début de la page *HELMo Learn* du cours. Les autres classes doivent être créées par vos soins. La fonction **main** doit être déclarée dans la classe **SautHauteur**.

Le projet déposé sur l'espace de cours doit être complet et exécutable !

FONCTIONNALITÉS INDISPENSABLES


Les fonctionnalités suivantes doivent être opérationnelles, tout en veillant à respecter les consignes stipulées dans cet énoncé :

- La fonction `ajouterElement` de la classe `TableauEntiers` doit être correctement réalisée, testée et utilisée.
- Les fonctions `ajouterElement` et `ajouterLigne` de la classe `TableauCaracteres` doivent être correctement réalisées, testées et utilisées.
- Dans la fonction `main` de la classe `SautHauteur`, un tableau de chaînes d'entiers à une dimension doit être correctement créé et initialisé (voir la section « Les structures de données ») afin d'y répertorier les **hauteurs des sauts exprimées en centimètres**.
- Dans la fonction `main` de la classe `SautHauteur`, un tableau de caractères à deux dimensions doit être correctement créé et initialisé (voir la section « Les structures de données ») afin d'y répertorier les **indicateurs de réussite/échec des sauts**.
- Dans sa version minimale, la fonction `main` doit permettre d'encoder aux moins 3 hauteurs sautées et les indicateurs de réussites/échecs correspondants, et ce, sans nécessairement les afficher en retour et devoir déterminer la fin de l'épreuve (voir l'exemple d'exécution « MINIMUM REQUIS » en fin d'énoncé). Ces données doivent être ajoutées aux deux tableaux cités aux points précédents en respectant rigoureusement les consignes (voir la section « Les structures de données »).

Dans la version minimale, il n'est pas nécessaire de vérifier la validité des données saisies par l'utilisateur.

Si l'une de ces fonctionnalités n'est pas observée, cela entraîne automatiquement l'échec pour l'examen. Attention, la réussite de ces fonctionnalités seules ne garantit pas l'obtention d'une note supérieure ou égale à 10 / 20 ! Ne vous contentez pas de ces fonctionnalités et veillez à soigner les tests unitaires et la javadoc des fonctions.

APPROCHE RECOMMANDÉE

Dans un premier temps, **ne réalisez que les fonctions indispensables**. Ces dernières sont indiquées dans la section « *Fonctionnalités indispensables* » et sont marquées d'un point d'exclamation  au niveau de leurs descriptions.

Réalisez ensuite la fonction **main** sur base des directives données dans la section « *Fonctionnalités indispensables* » et de l'exemple d'exécution **MINIMUM REQUIS**.

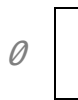
Il est recommandé de ne réaliser la suite du programme que lorsque ces deux premières étapes sont terminées.

LES STRUCTURES DE DONNÉES


ENREGISTRER LES HAUTEURS DES SAUTS

Utilisez un **tableau d'entiers à une dimension** pour y enregistrer les hauteurs des sauts exprimées en centimètres.

Initialement, ce tableau doit être vide :



Supposons maintenant que trois hauteurs différentes aient été sautées. Dans ce cas, un tableau semblable au suivant est obtenu :

	229	231	233
---	-----	-----	-----

*A chaque fois qu'une hauteur de saut supérieure à la précédente est spécifiée, la nouvelle hauteur doit être ajoutée au tableau à une dimension à l'aide de la fonction **TableauEntiers.ajouterElement**.*

ENREGISTRER LES INDICATEURS DE RÉUSSITE/ÉCHEC DES SAUTS

Utilisez un **tableau de caractères à deux dimensions** pour enregistrer les indicateurs de réussite/échec des sauts.

Initialement, ce tableau doit être vide (0 ligne).

Supposons maintenant que trois hauteurs différentes aient été sautées. Dans ce cas, un tableau semblable au suivant est obtenu :

0	'X'	'X'	'O'
1	'O'		
2	'X'	'X'	

Un caractère **'X'** majuscule (comme *Xavier*) indique un saut raté, tandis qu'un caractère **'O'** majuscule (comme *Olivia*) indique un saut réussi.

Chaque ligne du tableau à deux dimensions doit correspondre à une hauteur du tableau répertoriant les hauteurs des sauts. Par exemple, en considérant les exemples donnés à la page précédente et à celle-ci : la ligne d'indice 0 [**'X'**, **'X'**, **'O'**] correspond à la hauteur 229 située à l'indice 0, la ligne d'indice 1 [**'O'**] correspond à la hauteur 231 située à l'indice 1, et la ligne d'indice 2 [**'X'**, **'X'**] correspond à la hauteur 233 située à l'indice 2.

*A chaque fois qu'une hauteur de saut supérieure à la précédente est spécifiée, une nouvelle ligne vide doit être ajoutée au tableau à deux dimensions à l'aide de la fonction **TableauCaracteres.ajouterLigne**.*

*A chaque fois qu'une hauteur de saut supérieure ou égale à la précédente est spécifiée, le nouvel indicateur doit être ajouté à la dernière ligne du tableau à deux dimensions à l'aide de la fonction **TableauCaracteres.ajouterElement**.*

LES CLASSES TABLEAUENTIERS ET TABLEAUENTIERSTEST

L'unique fonction de la classe **TableauEntiers** doit être documentée avec un commentaire javadoc et testée à l'aide de JUnit 5 dans la classe **TableauEntiersTest**.



Dans la classe **TableauEntiers**, déclarez une fonction nommée **ajouterElement** qui retourne une copie des éléments d'un tableau d'entiers à une dimension augmentée d'un nouvel élément :

```
public static int[] ajouterElement(int[] t, int element)
```

Le paramètre **t** désigne le tableau dont les éléments doivent être copiés.

Le paramètre **element** désigne l'élément à ajouter à ceux du tableau **t**.

*Exemple : si **t** est le tableau [229, 231] et **element** l'entier 233, alors le tableau retourné est [229, 231, 233].*

[SUGGESTION] Aidez-vous de la fonction *copyOf* de la classe **Arrays**.

[PRÉCONDITION] La référence réceptionnée par le paramètre **t** est supposée valide (ne vaut pas **null**).

LES CLASSES TABLEAUCARACTERES ET TABLEAUCARACTERESTEST

Toutes les fonctions de la classe **TableauCaracteres** doivent être documentées avec des commentaires javadoc et testées à l'aide de JUnit 5 dans la classe **TableauCaracteresTest**.

Dans la classe **TableauCaracteres**, déclarez :



→ Une fonction nommée **ajouterElement** qui retourne une copie des éléments d'un tableau de caractères à une dimension augmentée d'un nouvel élément :

```
public static char[] ajouterElement(char[] t, char element)
```

Le paramètre **t** désigne le tableau dont les éléments doivent être copiés.

Le paramètre **element** désigne l'élément à ajouter à ceux du tableau **t**.

*Exemple : si **t** est le tableau ['X', 'X'] et **element** le caractère 'O', alors le*

tableau retourné est ['X', 'X', 'O'].

[SUGGESTION] Aidez-vous de la fonction `copyOf` de la classe `Arrays`.

[PRÉCONDITION] La référence réceptionnée par le paramètre `t` est supposée valide (ne vaut pas `null`).



→ Une fonction nommée `ajouterLigne` qui retourne une copie des références des lignes d'un tableau de caractères à deux dimensions augmentée de la référence d'une nouvelle ligne :

```
public static char[][] ajouterLigne(char[][] t, char[] ligne)
```

Le paramètre `t` désigne le tableau dont les références des lignes doivent être copiées.

Le paramètre `ligne` désigne la référence de ligne à ajouter à celles du tableau `t`.

Exemple : si `t` est le tableau [['X', 'X', 'O'], ['O']] et `ligne` le tableau ['X', 'X'], alors le tableau retourné est [['X', 'X', 'O'], ['O'], ['X', 'X']].

[SUGGESTION] Aidez-vous de la fonction `copyOf` de la classe `Arrays`.

[PRÉCONDITIONS] La référence réceptionnée par le paramètre `t` est supposée valide (ne vaut pas `null`). Cependant, le paramètre `ligne` peut recevoir n'importe quelle référence (la référence `null` ou d'un tableau vide est acceptée).

→ Une fonction nommée `enUneDimension` qui retourne la référence d'un tableau à une dimension contenant une copie des éléments d'un tableau à deux dimensions, ces éléments devant être copiés dans l'ordre de lecture « ligne par ligne » :

```
public static char[] enUneDimension(char[][] t)
```

Le paramètre `t` désigne le tableau contenant les éléments à copier dans le tableau à une dimension.

Exemple : si `t` est le tableau [['X', 'X', 'O'], ['O'], ['X', 'X']], alors la fonction retourne le tableau ['X', 'X', 'O', 'O', 'X', 'X'].

[CONTRAİNTE] Pour la réalisation de cette fonction, vous devez utiliser la fonction *ajouterElement* de la classe **TableauCaracteres** afin d'ajouter un à un les éléments au tableau à retourner. Initialement, ce tableau doit être vide.

[PRÉCONDITION] La référence réceptionnée par le paramètre **t** est supposée valide (ne vaut pas **null**).

[POSTCONDITION] Si **t** désigne un tableau vide, alors le tableau retourné est vide.

COMMENT UTILISER CETTE FONCTION DANS LE PROGRAMME PRINCIPAL ?

Cette fonction permettra de transformer le tableau répertoriant les indicateurs de réussite/échec en un tableau à une dimension afin de faciliter le comptage des sauts échoués consécutifs.

→ Une fonction nommée **dernierePositionDe** qui détermine la position de la dernière apparition d'un caractère au sein d'un tableau à une dimension, et ce, en tenant compte de la casse (par exemple, le caractère 'x' est considéré comme différent du caractère 'X') :

```
public static int dernierePositionDe(char[] t, char valeur)
```

Le paramètre **t** désigne le tableau contenant les éléments parmi lesquels la recherche doit être effectuée.

Le paramètre **valeur** désigne le caractère à rechercher.

*Exemple : si **t** est le tableau ['X', 'X', '0', '0', 'X', 'X'] et **valeur** le caractère '0', alors la fonction retourne l'entier 3.*

[PRÉCONDITION] La référence réceptionnée par le paramètre **t** est supposée valide (ne vaut pas **null**).

[POSTCONDITION] Si aucun caractère du tableau **t** ne correspond à la valeur spécifiée, alors la fonction retourne -1.

COMMENT UTILISER CETTE FONCTION DANS LE PROGRAMME PRINCIPAL ?

Bien utilisée, cette fonction permettra de compter les sauts échoués successifs afin de déterminer si l'épreuve se termine pour l'athlète.

LA CLASSE SAUTHAUTEUR

Seule la fonction **afficherSauts** décrite ci-dessous doit être documentée avec des commentaires javadoc. Elle ne doit cependant pas être testée à l'aide de JUnit 5.

Dans la classe **SautHuteur**, déclarez une fonction nommée **afficherSauts** qui produit en console une vue d'ensemble des sauts effectués par un athlète.

```
public static void afficherSauts(int[] hauteursSautsEnCm,
                                char[][] indicateursParSaut)
```

Le paramètre **hauteursSautsEnCm** désigne le tableau répertoriant les hauteurs des sauts déjà effectués.

Le paramètre **indicateursParSaut** désigne le tableau répertoriant les indicateurs de réussite/échec correspondant aux hauteurs du tableau **hauteursSautsEnCm**.

Exemple : si **hauteursSautsEnCm** est le tableau **[229, 231, 233]** et **indicateursParSaut** le tableau **[['X', 'X', 'O'], ['O'], ['X', 'X']]**, alors la fonction produit l'affichage suivant en console :

```
2m29 : X X O
```

```
2m31 : O
```

```
2m33 : X X
```

[PRÉCONDITIONS] Les références réceptionnées par les paramètres **hauteursSautsEnCm** et **indicateursParSaut** sont supposées valides (ne valent pas **null**). De plus, les longueurs des deux tableaux désignés sont supposées identiques.

LES EXEMPLES D'EXÉCUTION

MINIMUM REQUIS

Hauteur du saut (en cm) ? 229
 Saut réussi : (o)ui, (n)on ? n
 Saut réussi : (o)ui, (n)on ? n
 Saut réussi : (o)ui, (n)on ? o

L'acquisition de l'indicateur de réussite est répétée au maximum 3 fois. Dès qu'un saut est réussi, la répétition s'arrête.

Hauteur du saut (en cm) ? 231
 Saut réussi : (o)ui, (n)on ? o

Lorsque l'encodage des données de 3 sauts est terminé, le programme s'arrête. Ces données doivent être organisées dans les 2 tableaux de la fonction **main**, conformément aux consignes !

Hauteur du saut (en cm) ? 233
 Saut réussi : (o)ui, (n)on ? n
 Saut réussi : (o)ui, (n)on ? n
 Saut réussi : (o)ui, (n)on ? o

VERSION IDÉALE

Hauteur du saut (en cm) ? 169
 Hauteur du saut (en cm) ? 229
 Saut réussi : (o)ui, (n)on ? n
 2m29 : X

La hauteur minimale tolérée est de 170 cm.

Hauteur du saut (en cm) ? 229
 Saut réussi : (o)ui, (n)on ? N
 2m29 : X X

Une lettre majuscule est considérée comme valide en réponse à cette question.

Hauteur du saut (en cm) ? 219
 Hauteur du saut (en cm) ? 229
 Saut réussi : (o)ui, (n)on ? o
 Saut réussi : (o)ui, (n)on ? o
 2m29 : X X O

Il n'est pas permis de diminuer la hauteur.

Il n'est pas permis de répondre à cette question autrement que par une lettre 'o' ou 'n' (minuscule ou majuscule).

Hauteur du saut (en cm) ? 229
 Hauteur du saut (en cm) ? 231
 Saut réussi : (o)ui, (n)on ? o
 2m29 : X X O
 2m31 : O

Cette hauteur étant réussie, il faut obligatoirement saisir une hauteur supérieure.

Hauteur du saut (en cm) ? 233
 Saut réussi : (o)ui, (n)on ? n
 2m29 : X X O
 2m31 : O

2m33 : X

Hauteur du saut (en cm) ? 233

Saut réussi : (o)ui, (n)on ? n

2m29 : X X O

2m31 : O

2m33 : X X

Hauteur du saut (en cm) ? 234

Saut réussi : (o)ui, (n)on ? o

2m29 : X X O

2m31 : O

2m33 : X X

2m34 : O

L'athlète fait l'impasse sur la hauteur 2m33.

Hauteur du saut (en cm) ? 237

Saut réussi : (o)ui, (n)on ? n

2m29 : X X O

2m31 : O

2m33 : X X

2m34 : O

2m37 : X

Hauteur du saut (en cm) ? 238

Saut réussi : (o)ui, (n)on ? n

2m29 : X X O

2m31 : O

2m33 : X X

2m34 : O

2m37 : X

2m38 : X

L'athlète fait l'impasse sur la hauteur 2m37.

Hauteur du saut (en cm) ? 238

Saut réussi : (o)ui, (n)on ? n

2m29 : X X O

2m31 : O

2m33 : X X

2m34 : O

2m37 : X

2m38 : X X

Après 3 sauts échoués successifs, l'athlète ne peut plus continuer l'épreuve.

L'épreuve est terminée pour cet athlète.