

TP n°10

Exceptions et fichiers

Pour rappel :

- Le CM sur les exceptions et les exceptions personnalisées a été réalisé et vous retrouverez les slides dans la section CM sur *Moodle*.
- Le CM sur les flux a été réalisé et vous retrouverez les slides, une fiche récapitulative et des exemples de lecture/écriture dans la section CM sur *Moodle*.

Pour approfondir vos connaissances, n'hésitez pas à consulter la page officielle : API Java 8 (<http://docs.oracle.com/javase/8/docs/api/>).



Essayez de répondre aux différentes questions, si vous n'êtes pas sûr de votre réponse n'hésitez pas à questionner votre intervenant de TP.

1 Attrapez-les tous !

Nous utilisons le programme Pokemon disponible sur *Moodle*.

1. Récupérez-le et compilez-le.
2. Des erreurs de compilations nous annoncent que certaines exceptions obligatoires ne sont pas gérées. Capturez-les toutes (les non-vérifiées aussi) !

Maintenant que votre programme compile, son fonctionnement ne vous paraît-il pas étonnant ? Réfléchissez au type de flux utilisé et regardez la documentation de l'API Java.

3. Expliquez ce qui se passe et pourquoi l'affichage obtenu est différent de ce qui a été entré au clavier.

2 Le jeu du pendu

Téléchargez la classe Pendu et sa classe de test. Testez et analysez ce code.

1. Dans quel cas le mot à deviner n'est pas valide ? Que se passe-t-il dans ce cas ?

Comme vous l'avez normalement remarqué, la classe utilise un mot par défaut. Nous souhaitons gérer ce problème avec une exception personnalisée. Pour cela n'hésitez pas à relire le CM Partie 9 - Exception suite.

2. Créez l'exception PenduException héritant de la classe Exception. Vous proposerez 2 constructeurs, l'un avec un message d'erreur et l'autre avec un message d'erreur et le mot invalide. Vous pourrez ainsi proposer un affichage contenant le mot invalide et vous permettrez l'accès au mot invalide depuis l'exception personnalisée.
3. Modifiez ensuite la classe Pendu et la classe TestPendu afin de lever, propager et traiter cette exception personnalisée. La gestion d'exception remplace la solution du mot par défaut.

3 Le jeu du pendu le retour

Afin de pouvoir jouer seul, nous souhaitons pouvoir utiliser notre jeu du pendu avec en entrée un fichier de mots dans lequel on prendra un mot aléatoirement.

La classe `ListeMots` est caractérisée par un nom de fichier (à partir duquel les mots seront chargés) et un tableau dynamique (`Vector`) de chaînes de caractères (dans lequel les mots seront stockés). Elle possède un unique constructeur par initialisation et deux méthodes :

- `motAleatoire` : qui renvoie un mot du tableau dynamique aléatoirement
- `ajouterMot` : qui ajoute au tableau dynamique et au fichier le mot passé en paramètre, si le mot est déjà présent il ne sera pas ajouté (pas de doublon)

1. Ecrivez la classe `ListeMots`
2. Écrivez une classe de test permettant de vérifier le bon fonctionnement de la classe `ListeMots`. Celle-ci devra demander le nom du fichier au clavier, ouvrir le fichier correspondant, demander l'ajout au clavier de quelques mots additionnels, puis afficher l'ensemble des mots à l'écran. Assurez-vous que votre programme produit la bonne liste de mots, incluant vos ajouts, entre deux ouvertures/fermetures de fichier.

Nous allons maintenant modifier notre classe `TestPendul`. Elle doit :

3. Créer une liste de mots à partir du fichier `dicoPendul.txt` disponible sur *Moodle*.
4. Si le fichier a pu être lu, elle proposera à l'utilisateur un menu pour jouer au pendu à partir d'un mot de la liste tiré aléatoirement ou en saisissant lui-même un mot. Si l'utilisateur choisit de saisir un mot, à la fin du jeu il sera ajouté à la liste et donc au fichier.
5. Si le fichier n'a pas pu être lu, elle proposera uniquement la saisie du mot et il ne pourra être ajouté au fichier.

4 Ecriture et lecture d'objet dans un fichier

Reprenez le package `personne` de la fin du TP9 (contenant les classes `Personne`, `Enseignant`, `Etudiant` et `Vacataire`) ainsi que la classe de test correspondante.

1. Modifiez la classe de test afin de créer un tableau dynamique de `Personne` contenant une `Personne`, un `Enseignant`, un `Etudiant` et un `Vacataire`.

Le but est maintenant d'écrire chaque objet du tableau dynamique dans un fichier.

2. Faites cela dans la classe de test, à la suite de la création du tableau dynamique, en vous inspirant de la fiche de cours sur les flux.

Ce fichier n'est pas lisible directement pour nous, nous allons le déchiffrer.

3. Toujours à la suite de la classe de test, lisez chaque objet du fichier pour le placer dans un nouveau tableau dynamique de `Personne`. Vous afficherez ce nouveau tableau et vous comparerez chaque objet entre le premier et le nouveau tableau pour vérifier s'ils sont bien identiques.