APPRENDRE DU VOCABULAIRE DANS UNE LANGUE ÉTRANGÈRE.

La finalité du projet, que vous réaliserez en binôme, est de créer et pouvoir utiliser une application vous permettant de vous faire réciter des listes de vocabulaire dans une langue étrangère. Votre objectif sera d'écrire un code lisible, commenté et le plus factorisé possible.

Pour cela, vous utiliserez les structures de données que vous jugerez nécessaires, ainsi que les notions d'héritage, d'interface, de classe abstraite, de package, de generics, ... Si une fonctionnalité vous semble nécessaire, ou simplement intéressante pour votre future utilisation, il vous est vivement conseillé de l'implémenter.

1. Fonctionnement de l'application

L'application commencera par un menu où l'utilisateur aura le choix entre les différentes options possibles. Vous pourrez ajouter à loisir autant d'options que cela vous semble nécessaire.

Trois de ces options auront obligatoirement pour but :

(1) d'afficher la liste de vocabulaire ;

(2) de faire réciter une liste de vocabulaire ;

(3) d'évaluer la connaissance d'une liste de vocabulaire.

Pour cela, vous implémenterez :

(a) une classe `Word`, qui définira un type `Word`, avec (dans un premier temps) comme variables d'instances un mot dans la langue d'origine et le mot traduit ;

(b) une classe `VocList`, qui contiendra la liste de mots sur laquelle l'application fonctionnera en mode `LearningMode` ;

(c) une classe `LearningMode`, avec comme variables d'instances une liste de vocabulaire (c'est-à-dire une instance de la classe `VocList`) ainsi que la chaine de caractères nommant la liste de vocabulaire en question ; en particulier, elle contiendra une méthode `use()` permettant de faire réciter cette liste de vocabulaire ;

(d) une classe `Grade`, qui contiendra toutes les méthodes nécessaires pour la gestion des notes en mode `ExamMode` ;

(e) une classe `GradedWord`, qui définira un type `GradedWord`, avec comme variables d'instances un `Word` et sa note courante de type `Grade` ;

(f) une classe `GradedVocList`, qui contiendra la liste de mots sur laquelle l'application fonctionnera ;

(g) une classe `ExamMode`, qui contiendra une méthode `use()` permettant d'évaluer la connaissance d'une liste de vocabulaire de type `GradedVocList`.

Une fois que l'utilisateur décidera d'arrêter l'apprentissage de la liste courante de vocabulaire ou une fois l'évaluation finie, l'application reviendra sur le menu de base (c'est-à-dire le premier menu obtenu au lancement de l'application), où il pourra choisir d'éteindre l'application, de charger une autre liste de vocabulaire, ...

La création et le chargement d'une liste de vocabulaire s'effectuera par l'intermédiaire de fichiers `.vocab`. La création s'effectuera en lisant le contenu d'un fichier `.txt` créé par l'un des utilisateurs (selon un calibre que vous choisirez) ; la sauvegarde s'effectuera en implémentant l'interface `Serializable` pour chaque classe en rapport avec les variables d'instances de `VocList`. Ces actions seront implémentées dans la classe `FileVoc`. Dans un premier temps, l'utilisateur écrira lui-même le mot à traduire ainsi que sa traduction dans les fichiers `.txt`. Plus tard, et seulement plus tard, l'utilisation d'un dictionnaire automatique (via internet) pourra être utilisé pour n'avoir à donner que le mot dans la langue d'origine (cf. niveau C2).

2. Consignes et rendus

Vous effectuerez votre projet en binôme ; les binômes seront constitués avant le 04/11/13. Vous rendrez votre projet par courrier électronique avant le 27/11/13. L'objet devra impérativement être :

"[EISC2] - Projet de `Prenom NomDeFamille1` et `Prenom NomDeFamille2`."

Votre courrier electronique contiendra en pièce jointe une archive contenant vos fichiers `.java`, `.vocab`, `.txt` ayant servi à la création de vos listes de vocabulaire et un fichier de compte-rendu de projet. Pour mémoire, un courrier electronique contient aussi un corps de mail.

Votre compte-rendu devra suivre la structure suivante :

I. expliquer en détails le fonctionnement de votre application à travers les différentes options que vous avez implémentées, ainsi que le calibre utilisé pour lire les `.txt` nécessaire à la création des `.vocab` ;

II. rendre intelligible vos choix d'implémentation : en particulier, pour chaque classe, vous expliquerez vos imports, vous indiquerez les raisons qui vous ont poussées à faire hériter une classe d'une autre, à déclarer un interface et à implémenter tel ou tel interface, à utiliser la notion de generics, ... ; pour chaque variable d'instance, vous expliquerez aussi succintement votre choix de structure de données ainsi que les choix des différents modifiers qui s'y rapportent ;

III. indiquer les difficultés que vous avez rencontrées et les différentes solutions mises en oeuvre pour les contourner (si vous aviez le choix entre plusieurs solutions, vous indiquerez aussi les raisons de votre choix ; si vous avez recopié du code et ne comprenez pas parfaitement son fonctionnement, c'est ici qu'il vous faudra le mentionner, ...) ; en cas d'absence de solutions, vous l'indiquerez également ;

IV. détailler les options personnelles de fonctionnement que vous avez implémentées, ainsi que celles que vous auriez aimé implémenter si le temps vous l'avez permis.

Enfin, vos différents fichiers `.java` devront être commentés en anglais ; vos noms de classes, de méthodes et de variables seront aussi choisis en anglais et devront être en rapport direct et compréhensible avec leur signification réelle.

La notation du projet prendra autant en compte votre compte-rendu (notamment votre honnêteté quant à la section sur les difficultées rencontrées...), la qualité du code et des commentaires, que la soutenance de projet que vous réaliserez. Cette dernière durera au maximum 15 minutes et sera prévu entre le 02/12/13 et le 05/12/13. Son but sera de montrer le fonctionnement de votre application, d'expliquer vos choix d'implémentation, ... Vous pourrez alors suivre votre compte-rendu.

Pour information, le plagiat sera très sévèrement sanctionné !...

3. Différents niveaux de réalisation de l'application

Votre objectif devra être d'implémenter l'application décrite ci-dessus et complétée selon votre envie. Six niveaux de complexité vous sont proposés pour vous guider, même si au final le niveau B1 sera requis pour le rendu de votre projet.

Pour chacun de ces niveaux, il s'agit de faire au moins ce qui est proposé ; il sera aussi particulièrement apprécié que vous complétiez cette liste d'items par vos propres idées à partir du niveau B1. Les niveaux C1 et C2 pourront être fait dans n'importe quel ordre, étant donnée qu'ils sont de complexité équivalente.

3.1. **Niveau A1 - "Bryan is in the kitchen".**

- Implémenter les classes `Word`, `VocList` et `LearningMode` ;
- Implémenter une classe `Menu` correspondant au fonctionnement en console décris au paragraphe 1 ;
- Implémenter une classe `Main` pour tester le fonctionnement de votre application en mode `LearningMode`.

3.2. **Niveau A2 - "It was the simplicity itself".**

- Implémenter les classes `Grade`, `GradedWord`, `GradedVocList` et `ExamMode` ;
- Modifier ce qu'il y a à modifier dans les classes `Grade`, `Word`, `GradedWord`, `VocList`, `GradedVocList`, `LearningMode` et `ExamMode` pour factoriser au maximum votre code (n'hésitez pas à utiliser autant que possible les notions d'héritages, d'interface, de classe abstraite ou à paramétrer un type si nécessaire) ;
- Implémenter la possibilité de fusionner plusieurs listes de vocabulaire lorsque l'utilisateur a validé leur connaissance en mode exam ;
- Implémenter une classe `Menu` correspondant à un fonctionnement en console, de sorte que votre menu indique d'une manière ou d'une autre à l'utilisateur quelles listes de vocabulaire peuvent être lues, utilisées ou fusionnées.
- Implémenter une classe `Main` pour tester le fonctionnement de votre application en mode `LearningMode` ou en mode `ExamMode`.

3.3. **Niveau B1 - "It's raining cats and dogs".**

- Avoir réalisé tous les items du niveau A2 ;
- Compléter l'implémentation de la classe `Word` afin de prendre en compte la nature des mots (c'est-à-dire indiquer s'il s'agit de noms, d'adjectifs, de verbes, ...) ; si le mot demandé à l'utilisateur est un mot irrégulier (par exemple un verbe ou un adjectif irrégulier en anglais, comme `bad worse worst`), l'utilisateur devra alors indiquer la totalité des champs nécessaires lors de l'utilisation de l'application (dans l'exemple précédent, l'utilisateur devra donc écrire : `bad worse worst`).

Bonus : Implémenter toutes les options supplémentaires qui vous semble utile à votre propre utilisation de l'application.

3.4. **Niveau B2 - "There was a little witch which switched from Chicester to Ipswich".**

- Avoir réalisé tous les items du niveau B1 ;
- Implémenter une classe `Synonym` et les options correspondantes dans le menu pour que l'utilisateur puisse avoir aussi accès à des listes de synonymes dans les modes `LearningMode` et `ExamMode` ; toutes les modifications nécessaires seront faites dans le code.

Bonus : Implémenter toutes les variantes de la classe `Synonym` qui peuvent vous sembler utile à votre propre utilisation de l'application.

3.5. **Niveau C1 - "It's gonna be legend-... wait for it... dairy !"**

- Avoir réalisé tous les items du niveau B2 ;
- Implémenter une interface graphique à l'aide de `swing` pour que l'utilisation de l'application ne se fasse plus en console.

Si votre interface graphique fonctionne (correctement ou partiellement), lors de votre rendu, vous joindrez également à votre archive les fichiers permettant de faire fonctionner votre application en console. Pour cela, vous aurez la possibilité de modifier certains noms de classes judicieusement.

3.6. **Niveau C2 - "I have a dream".**

- Avoir réalisé tous les items du niveau C1 ;
- Implémenter une classe `AutomaticTranslation` qui permettra à l'utilisateur de n'avoir à indiquer dans les listes de vocabulaire `.txt` que le mot ou l'expression dans le langage d'origine ; pour cela vous accéderez au code source `HTML` d'un traducteur automatique sur internet et vous écrirez un parser sur ce code source pour obtenir directement la traduction demandée.

Pour ce niveau, l'application pourra, selon que vous ayez réalisé le niveau C1 ou non, fonctionner aussi bien en console qu'à l'aide d'une interface graphique.

* *

*

Pour mémoire, vous pouvez me contacter par mail en cas de difficultés, prendre rendez vous :

mail : olivier.bouillot@univ-mlv.fr
bureau : 4B03R

Votre calendrier est :

04/11/13	Constitution des binômes.
27/11/13	Rendu par mail de votre projet.
Entre le 02/12/13 et le 05/12/13	Soutenance de projet de 15 minutes maximum.

Une ou deux courtes séances de questions pourront être mises en oeuvre si besoin.

GOOD LUCK!

Et... N'oubliez pas ce que le sage Albert Einstein disait :

"Anyone who has never made a mistake has never tried anything new."