

But du laboratoire (2 heures)

L'objectif de ce laboratoire est de consolider l'utilisation des différents éléments de programmation vus jusqu'à maintenant. Pour ce faire, vous allez écrire un programme permettant à un utilisateur de deviner un nombre secret.

Toutes les données de ce laboratoire se trouvent sur le site web du cours <http://inf1.begincoding.net> dans le thème *Laboratoires*.

Partie 1 - L'application SecretNumber

Au lancement de l'application **SecretNumber**, le menu principal offre à l'utilisateur la possibilité de choisir la taille du nombre à deviner entre 1 et 10, entre 1 et 100, entre 1 et 1'000 et entre 1 et 10'000 (options 1 - 4).

Ensuite, l'application tire un nombre au hasard dans la plage de valeurs choisie. L'utilisateur peut maintenant proposer des nombres que l'application acquitte avec des messages indiquant si le nombre à deviner est plus grand ou plus petit. L'utilisateur continue à proposer des nombres jusqu'à ce que le bon nombre est deviné. Alors un message de félicitation s'affiche et l'utilisateur peut indiquer s'il veut recommencer le jeu ou non.

Ci-dessous vous voyez un déroulement possible de l'application :

```
*****
* Secret Number - Can you find it ? *
*****
Choose difficulty level :

      (1) Rookie level      (1 to 10)
      (2) Easy level       (1 to 100)
      (3) Normal level     (1 to 1000)
      (4) Nightmare level  (1 to 100000)

>> What is your choice ? 1
>> Enter your guess : 6
The number is bigger than 6
>> Enter your guess : 9
The number is bigger than 9
>> Enter your guess : 10

!!!! Congratulations !!!!
```

Tâche 1

1. Créez un nouveau projet *Eclipse* pour ce labo.
2. Ajoutez une nouvelle classe `SecretNumber` à votre projet. Écrivez une procédure `displayMenu()` qui affiche les quatre options du menu à l'écran avec le petit titre que l'on peut voir sur l'exemple.
3. Ajouter ensuite une fonction `askUser` qui prend en paramètre un `String` contenant une question et qui retourne un entier. Cette fonction doit afficher la question puis retourner la valeur lue au clavier à l'aide de la méthode `Input.readInt()` fournie.
4. Si le choix est valide, la valeur de la borne supérieure de la plage de valeurs choisie est mémorisée dans la variable `upperLimit`. Sinon, le menu est réaffiché et l'utilisateur est prié de faire son choix. Avant de quitter le programme, le contenu de la variable `upperLimit` doit être affiché à l'écran.
5. Testez votre programme.

Tâche 2

1. Écrivez une fonction `generateSecretNumber` prenant en argument la limite du nombre que l'on veut et qui retourne un entier compris entre 1 et la limite. Pour ce faire, utilisez la fonction `Math.random()` qui retourne un nombre compris entre [0 et 1[ainsi que la fonction `Math.floor(x)` qui arrondit vers le bas un nombre passé en argument.
2. Utilisez `generateSecretNumber` afin de stocker dans la variable `secretNumber` le nombre à trouver.
3. Testez votre programme et vérifiez le contenu de la variable `secretNumber`.

Tâche 3

1. Complétez votre programme avec le code qui demande à l'utilisateur d'introduire un nombre. Ensuite le programme indique si le nombre à deviner est plus grand ou plus petit que le nombre introduit. Ceci est répété jusqu'à ce que le bon nombre a été introduit. Dans ce cas, un message de félicitations est affiché et le programme se termine.
2. Complétez le programme afin que l'utilisateur puisse dire s'il veut rejouer une partie ou non. Si oui, le programme reprend avec l'affichage du menu principal. Sinon, le programme se termine avec un message d'adieu.
3. Vous pouvez au besoin utiliser la méthode `System.exit(int x)` qui permet de quitter immédiatement le programme exécuté (l'argument permet de spécifier l'état du programme à la fin de l'exécution. Normalement on utilise 1 si tout va bien et -1 si le programme s'est terminé en état d'erreur).

Tâche 4

1. Complétez le programme afin que le nombre d'essais soit affiché au moment où le nombre correct a été deviné.
2. Complétez le programme pour afficher un message de félicitations lorsque le nombre d'essais est inférieur à 3.
3. Complétez le programme pour que l'utilisateur puisse entrer le nombre 0 pour abandonner la partie en cours. Dans ce cas, la bonne solution est affichée à la place du message de félicitations.

Tâche 5

Le but de cette tâche est d'implémenter une recherche dichotomique (explications en classe et <http://fr.wikipedia.org/wiki/dichotomie>) pour trouver le nombre aléatoire qui a été sélectionné dans votre code.

1. Ajoutez une fonction qui calcule le nombre d'itérations nécessaires pour trouver le nombre cherché à l'aide d'une recherche dichotomique.
2. Complétez le programme afin que le nombre d'itérations nécessaires soit inscrit en dessous du nombre d'essais et affichez de nouvelles félicitations si l'utilisateur a trouvé le nombre plus rapidement.