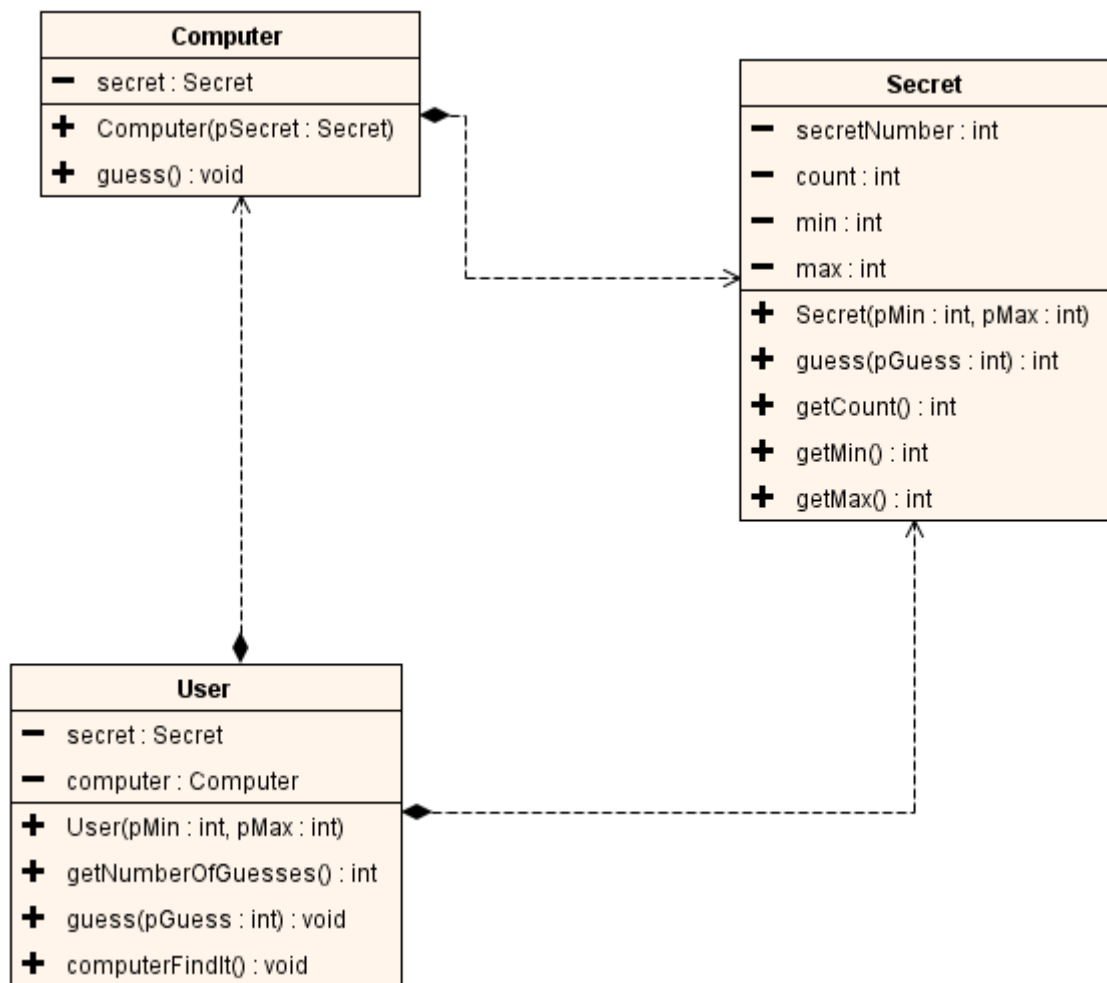


Guess my number

Vous devez implémenter le projet **Guess my number**. Il s'agit d'un jeu où l'utilisateur doit deviner un nombre que l'ordinateur a généré. En plus, il est possible que l'ordinateur devine le nombre (sans tricher) en employant une stratégie efficace.

Il est important de respecter le diagramme UML suivant :



Classe Secret

1. Attributs et manipulateurs (pas de manipulateur pour `secretNumber`)
 - `secretNumber` contient un nombre entier entre `[min ; max]`
 - `count` contient le nombre de fois que l'utilisateur et/ou l'ordinateur ont deviné un nombre
 - `min` et `max` définissent l'intervalle dans lequel se trouve `secretNumber`
2. Le constructeur initialise `min` et `max` avec ses paramètres respectifs. Ensuite, il programme `secretNumber` avec un nombre aléatoire entier compris dans l'intervalle `[min ; max]`. `count` est initialisé à 0.
3. La méthode `guess(...)` incrémente `count`. Ensuite, il compare `pGuess` à `secretNumber`.
Il en résulte 3 possibilités :
 1. Si `pGuess` est égal à `secretNumber`, la valeur 0 est retournée.
 2. Si `pGuess` est plus grand que `secretNumber`, la valeur 1 est retournée.
 3. Si `pGuess` est plus petit que `secretNumber`, la valeur -1 est retournée.

Classe Computer

1. L'attribut `secret` contient une référence vers une instance de la classe `Secret`.
2. Le constructeur initialise `secret` avec son paramètre `pSecret`.
3. La méthode `guess()` cherche de façon systématique le nombre secret. Pour cela, vous devez implémenter une stratégie qui vous permet de rétrécir l'intervalle des possibilités.

Classe User

1. Attributs
 - L'attribut `secret` contient une référence vers une instance de la classe `Secret`.
 - L'attribut `computer` contient une référence vers une instance de la classe `Computer`.
2. Le constructeur crée une instance de la classe `Secret` à l'aide de ses paramètres et enregistre la référence de cet objet dans `secret`. Ensuite, il crée une instance de la classe `Computer` en passant la référence stockée dans `secret` et enregistre la référence de cet objet dans `computer`.
3. `getNumberOfGuesses()` retourne le nombre de fois que l'utilisateur et/ou l'ordinateur ont deviné un nombre. Utilisez l'attribut `secret` pour déterminer ce nombre.
4. `guess(...)` demande à l'instance de `Secret` si la valeur contenue dans le paramètre est égal au nombre secret. Ensuite, en fonction du résultat un des 3 messages possibles est affiché à la console:
 - "Great. I guessed the right number"
 - "My guess was too small"
 - "My guess was too high"
5. `computerFindIt(...)` fait appel de façon répétée à la méthode `guess()` de l'instance de la classe `Computer`. Lorsque le nombre secret est trouvé, le nombre d'essais est affiché à la console, exactement comme montré dans l'exemple suivant :

Total number of guesses: 16