

Exercice 4 : Écrire une classe `JoueurHumain` pour encapsuler les entrées-sorties du joueur humain. Définir les méthodes suivantes :

- `public int getCoup()` : demande au joueur son coup et le renvoie.
- `public void setReponse(int reponse)` : affiche la réponse.
- `public int testSecret(int essai)` : affiche l'essai puis demande la réponse du joueur et la renvoie.

Modifier le programme afin d'utiliser un objet de la classe `JoueurHumain` en opposition au programme.

Exercice 5 : Écrire une classe `JoueurIncremental` qui effectue une recherche incrémentale (0, puis 1, etc.) du secret. Définir les méthodes suivantes :

- `public int getCoup()` : renvoie le prochain coup joué par ce joueur.
- `public int testSecret(int essai)` : renvoie 0 si c'est le bon nombre, 1 s'il est trop petit et 2 s'il est trop grand.

La borne supérieure de l'intervalle de recherche sera passée en paramètre du constructeur.

Modifier le programme afin d'utiliser un objet de la classe `JoueurIncremental` en opposition à un objet de la classe `JoueurHumain`.

Exercice 6 : Définir une classe `JoueurDichotomique` qui cherche le secret en effectuant une recherche dichotomique.

1. Pour cela, définir une classe `Intervalle` permettant de représenter l'intervalle de recherche, avec les méthodes suivantes :

- `void setMin(int m)` : fixe le nouveau minimum.
- `void setMax(int m)` : fixe le nouveau maximum.
- `int getBorneInf()` : renvoie la borne inf.
- `int getBorneSup()` : renvoie la borne sup.
- `int getMilieu()` : renvoie le milieu de l'intervalle.
- `boolean estVide()` : renvoie vrai si l'intervalle est vide.

2. Définir les méthodes suivantes :

- `public int getCoup()` : renvoie le prochain coup joué par ce joueur.
- `public void setReponse(int reponse)` : indique la réponse pour la mise à jour de l'intervalle de recherche.
- `public int testSecret(int essai)` : renvoie 0 si c'est le bon nombre, 1 s'il est trop petit et 2 s'il est trop grand.

La borne supérieure de l'intervalle de recherche sera passée en paramètre du constructeur.

3. Modifier le programme afin d'utiliser un objet de la classe `JoueurDichotomique` en opposition à un objet de la classe `JoueurHumain`.

Exercice 7 : Définir une classe `Arbitre` qui fait jouer deux joueurs en vérifiant qu'aucun des deux ne triche. Il prendra en paramètre la valeur maximale de l'intervalle de jeu, un `JoueurIncremental` et un `JoueurHumain`. Il utilisera deux objets de la classe `Intervalle` pour vérifier la validité des réponses et interrompra la partie en cas de triche, en affichant un message indiquant qui a triché. La méthode réalisant la partie aura le prototype suivant :

- `public void jouer()`.