

SEMAINE 1 : TP2 - UTILISER DES TYPES OBJETS

1 EDITER, COMPILER, INTERPRÉTER UN PROGRAMME JAVA

On reprend l'exemple du cours.

Le code des classes **Joueur** et **JeuToutSeul** est fourni.

```
/**
 * La classe Joueur modélise un joueur de dés avec son nom, son cumul de points et son nombre de lancers
 */
public class Joueur {

    /**
     * Nom du joueur
     */
    String nom ;

    /**
     * Cumul du joueur
     */
    int cumul ;

    /**
     * Nombre de lancers du joueur
     */
    int nbLancers ;

    /**
     * Générateur valeurs aléatoires
     */
    java.util.Random r=new java.util.Random();

    /**
     * Crée un Joueur avec un nom n, cumul et nombre de lancers à 0
     * @param n le nom du joueur
     */
    Joueur(String n){
        nom=n;
    }

    /**
     * Méthode ajoutant au cumul le résultat d'un lancer de dé
     */
    void jouer() {
        cumul += r.nextInt(6)+1;
        nbLancers++;
    }

    /**
     * Compare le cumul de deux joueurs
     * @param j le joueur à comparer
     * @return vrai si le cumul du joueur courant est plus grand que celui de j
     */
    boolean bat(Joueur j){
        return nbLancers<j.nbLancers;
    }

    /**
     * Retourne une chaîne représentant l'état d'un Joueur
     * @return la chaîne représentant l'état d'un Joueur
     */
    public String toString() {
        return "Joueur [nom=" + nom + ", nbLancers=" + nbLancers + ", cumul=" + cumul + "]";
    }
}
```

```
/**
 * La classe JeuToutSeul modélise le scénario où "César" joue tout seul au "Jeu de dés"
 */
public class JeuToutSeul {

    public static void main(String[] args) {
        //Initialisation du total à atteindre
        int total=20;

        //Création d'un Joueur appelé César
        Joueur cesar=new Joueur("César");

        //On fait jouer César jusqu'à ce que il ait atteint le total demandé
        while(cesar.cumul<total) {
            cesar.jouer();
        }

        //Affichage du nombre de lancers nécessaires pour que César atteigne le total voulu
        System.out.println(cesar.nblancers);
    }
}
```

JOUEUR - JEUTOUTSEUL

Créer les classes Joueur et JeuToutSeul

Créer un répertoire **TP2** contenant les répertoires **src** et **bin**.

Copier les classes ci-dessus en nommant les fichiers contenant les programmes respectivement **Joueur.java** et **JeuToutSeul.java**.

Compiler et exécuter la classe **JeuToutSeul**.

Pour la compilation, si vous vous trouvez dans le répertoire **TP2** :

```
javac -sourcepath src -d bin JeuToutSeul.java
```

On remarque que **Joueur.java** est automatiquement compilé (vous pouvez le vérifier en regardant dans le répertoire **bin**).

Pour l'exécution, toujours en étant dans **TP2** :

```
java -cp bin JeuToutSeul
```

Questions :

1. Que représente l'affichage?
2. Change-t-il si on exécute à nouveau l'application?
3. Expliquer.

JOUEURBIS - JEUTOUTSEULBIS

On souhaite à présent développer une variante du jeu précédent :

- le joueur démarre avec un cumul égal à 20,
- la valeur du dé est retirée à chaque lancer,
- le jeu s'arrête lorsque le cumul est inférieur à 0.

Ecrire les classes **JoueurBis** et **JeuToutSeulBis**.

JOUEUR - JEUDEUX

A deux c'est mieux que tout seul...

Ecrire une classe **JeuADeux** qui fait jouer deux joueurs de type **Joueur** et affiche le nom du joueur gagnant.

Les joueurs jouent successivement jusqu'à atteindre le total.

Le gagnant est celui qui a effectué le moins de lancers.

JOUEUR - JEUAPLUSIEURS

Plus on est de fous ...

Ecrire une classe **JeuAPlusieurs** qui fait jouer un nombre quelconque de joueurs de type **Joueur**.

Les joueurs jouent successivement jusqu'à atteindre le total.

Les gagnants sont ceux qui ont effectué le moins de lancers.

Les noms des gagnants sont affichés (*il peut y avoir des ex-aequo*).

Le nombre de joueurs souhaité est fixé au début de la méthode **main()**.

On pourra utiliser un tableau de **Joueur** pour le traitement.

2 REUTILISER UNE CLASSE JAVA

SAISIE

On fournit une classe **Saisie** pour interagir avec l'utilisateur.

La classe **Saisie** permet de *recupérer* les valeurs saisies par l'utilisateur, et éventuellement de vérifier que la valeur saisie est un entier.

La documentation de cette classe, sa **javadoc**, est fournie sur Moodle.

Lire la **javadoc** !!!

Modifier la classe **JeuAPlusieurs** pour faire en sorte que le nombre de joueurs souhaité, ainsi que le total à atteindre, soient saisis par l'utilisateur.

*Etonnamment la saisie des valeurs s'effectue grâce à la classe **Saisie** fournie !*

Télécharger la classe **Saisie.class** depuis Moodle et la placer dans le répertoire **bin**.

GAGNANT GAGNANT

On raffine la définition de "gagnant", les gagnants sont ceux qui ont utilisé le moins de lancers, et qui ont la valeur cumulée la plus élevée.

Compléter la classe **JeuAPlusieurs** pour n'afficher que ces gagnants.

SAISIE DES NOMS

Compléter la classe **JeuAPlusieurs** pour permettre à l'utilisateur de saisir les noms des **Joueur**.

Pour éviter les homonymes, on s'assure que le nom du joueur saisi n'est pas déjà utilisé.

La méthode **equals()** de la classe **java.lang.String** permet de comparer deux chaînes de caractères.

POUR ALLER PLUS LOIN

Modifier le jeu pour que les joueurs lancent le dé chacun leur tour.