

TP N°3.2

INSTANCES ET METHODES

Objectif du TP

L'objectif de cet TP est de mettre en place des méthodes et de les utiliser.

Projet approche-objet

Exercice Cercle

- Dans le package **fr.diginamic.entites**, créez une classe **Cercle**
 - cette classe a un seul attribut d'instance : son rayon de type double
- Créez un constructeur pour cette classe avec le rayon en paramètre.
- Créez une méthode qui retourne le périmètre du cercle
- Créez une méthode qui retourne la surface du cercle.
- Dans le package **fr.diginamic.essais**, créez une classe **TestCercle**
 - Instanciez 2 cercles différents et affichez les résultats des méthodes de calcul de périmètre et de calcul de surface

Exercice CercleFactory

- Créez un package **fr.diginamic.utilis**
- Dans ce package créez une classe **CercleFactory**
 - cette classe a un méthode **de classe** qui prend en paramètre un double et retourne un Cercle
- dans la classe **TestCercle**, faites appel à la méthode static de **CercleFactory** pour créer vos cercles.
- **PRECISION** : en POO, on appelle **Factory** une classe qui en construit une autre. Il existe aussi le concept de Builder.

Exercice Operations

- Créez un package **fr.diginamic.operations**
- Dans ce package créez une classe **Operations**
 - cette classe a une méthode static **calcul** qui prend en paramètre 2 double a et b et un opérateur qui est de type char.
 - Si l'opérateur vaut '+' alors la méthode **calcul** retourne a+b

- Si l'opérateur vaut '-' alors la méthode **calcul** retourne a-b
- Faites la même chose pour les opérateurs * et /
- Créez une classe **TestOperations** qui permet de tester les 4 opérations.

Exercice CalculMoyenne

- Dans le package **fr.diginamic.operations** créez une classe **CalculMoyenne**
 - Cette classe a un **attribut d'instance** de type tableau de double.
 - Le constructeur de la classe **CalculMoyenne** a un paramètre de type entier qui permet de définir la longueur tableau.
 - Cette classe a également une **méthode ajout** qui permet d'ajouter un double au tableau. Pour réaliser cette méthode vous aurez peut-être besoin de gérer un index interne de la classe.
 - Créez une méthode **calcul** qui calcule et retourne la moyenne des éléments du tableau
- Créez une classe **TestMoyenne**
 - Vérifiez que votre classe **CalculMoyenne** correctement en effectuant au moins 2 tests différents.
 - Testez votre classe dans des cas « aux limites » :
 - que se passe t'il si j'instancie la classe **CalculMoyenne** avec une longueur négative
 - que se passe t'il si j'ajoute trop d'éléments à la classe ?
 - ajoutez des contrôles afin d'éviter les plantages à l'exécution.
 - ajoutez également des messages permettant d'indiquer à l'utilisateur que la/les méthode(s) sont incorrectement utilisée(s).
- **PRECISION** : c'est une bonne pratique d'avoir le réflexe de faire des classes utilitaires de ce type.

Exercice TrierTableau

- Dans le package **fr.diginamic.utils** créez une classe **TriTableau**
 - Cette classe a **méthode de classe** qui prend en paramètre un tableau et retourne un **tableau trié**.
 - Vous pouvez reprendre un algorithme écrit lors des TP approche impérative
 - Créez une classe qui permet de vérifier que la méthode fonctionne.
- **PRECISION** : c'est une bonne pratique d'avoir le réflexe de faire des classes utilitaires de ce type.

Commitez vos développements sur GitHub