

TP9-10

Conception par Contrat du package “visustats”

Objectif

L’objectif de cette série des TP9 et 10 est d’utiliser l’approche de Conception par Contrat vue en cours pour continuer le développement du package Java `fr.univrennes.istic.l2gen.visustats` dans l’objectif de représenter graphiquement des données statistiques, avec des diagrammes en colonne, diagrammes en barres, diagrammes camemberts, et plus tard diagrammes linéaires, ou encore diagramme de Kiviati. Ce package utilisera bien-sûr exclusivement votre package `fr.univrennes.istic.l2gen.geometrie` pour dessiner les formes primitives nécessaires.

Pendant ces TP vous aurez une évaluation individuelle de votre contribution au projet, et à la fin du TP8 chaque équipe devra présenter une démonstration scénarisée de son projet. La qualité du code et des tests de vos packages `geometrie` et `visustats` sera aussi évalué à la fin du TP8. Ceci constituera un élément de votre note de Contrôle Continu.

Renforcement du package `geometrie`

Implantation des contrats des `IFormes`

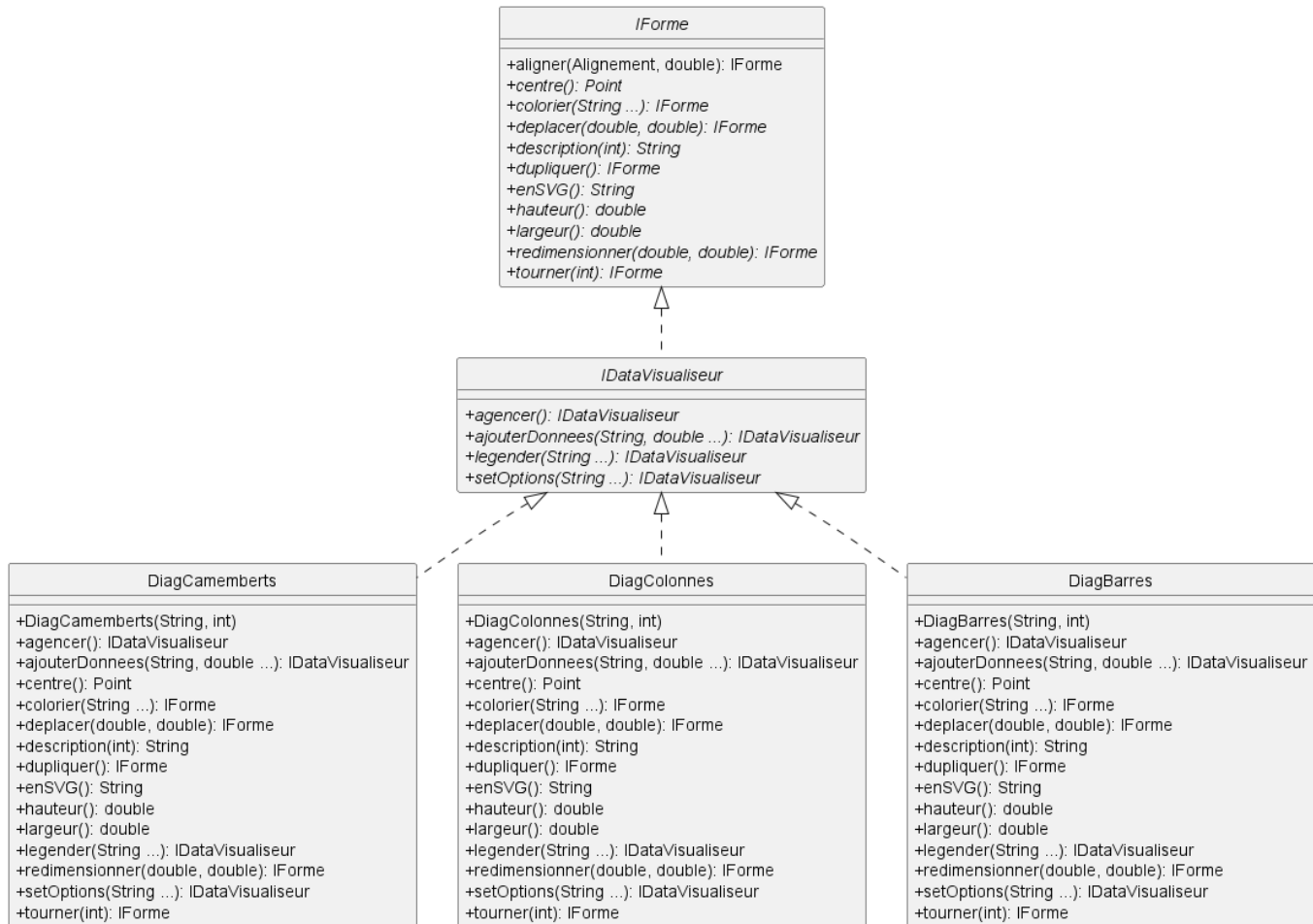
Spécifiez les contrats (préconditions et postconditions) des différentes méthodes de l’interface `IForme` tels que vus en cours. Intégrez-les aux commentaires Javadoc de `IForme`, puis répartissez-vous le travail pour écrire le code les vérifiant dans les différentes classes implémentant cette interface dans votre package `geometrie`.

Mise à jour des tests unitaires

Répartissez-vous le travail pour mettre à jour les tests unitaires de votre package `geometrie` tenant compte de ces contrats. En particulier ajouter les tests aux limites (par exemple `redimensionner(0,0)` et vérifier que la bonne exception est levée : `IllegalArgumentException`).

Application au développement du package `visustats`

Après la synchronisation habituelle de votre équipe sur votre GitLab, répartissez-vous le travail pour réaliser l’architecture ci-dessous, en commençant par spécifier les contrats de chacun des composants.



Utilisation

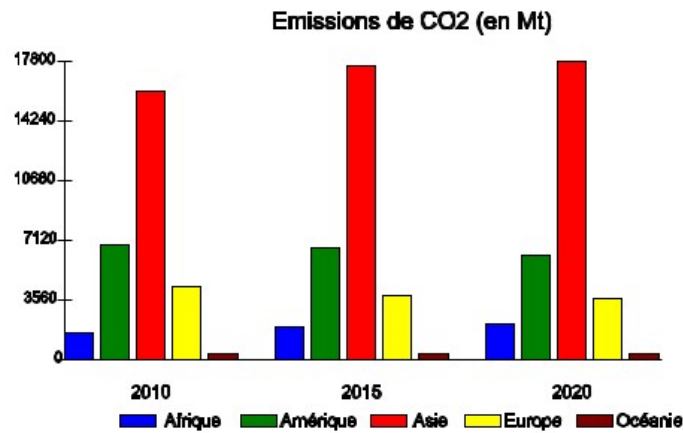
L'exécution du programme suivant :

```

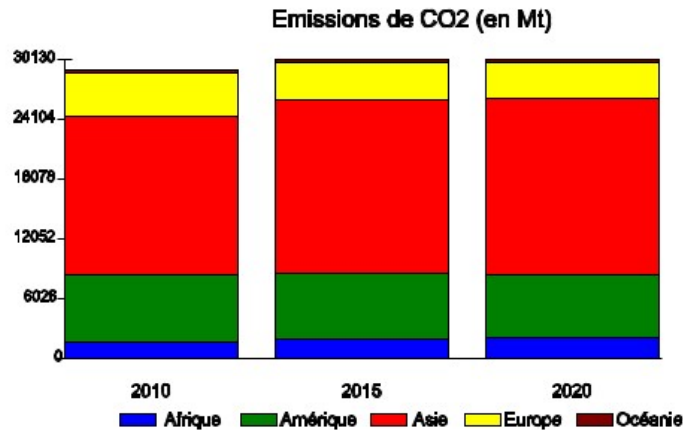
DiagColonnes visualiseur = new DiagColonnes("Emissions_de_CO2(en_Mt)");
visualiseur.legender("Afrique", "Amerique", "Asie", "Europe", "Oceanie");
visualiseur.ajouterDonnees("2010", 1600, 6800, 16000, 4300, 300);
visualiseur.ajouterDonnees("2015", 1900, 6600, 17500, 3800, 330);
visualiseur.ajouterDonnees("2020", 2100, 6200, 17800, 3600, 340);
visualiseur.colorier("Blue", "Green", "Red", "Yellow", "Maroon");
visualiseur.agencer().enSVG();

```

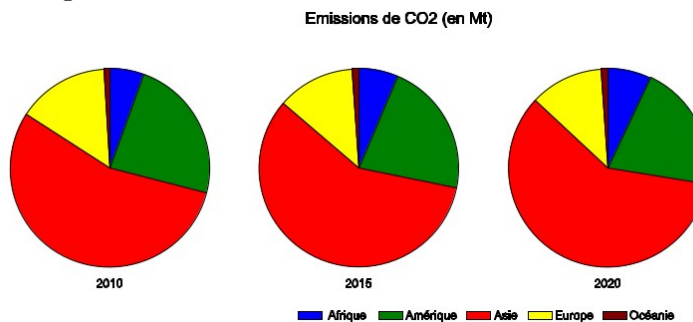
devrait produire une figure ressemblant à :



Et de la même manière pour DiagBarres :



Et bien sûr aussi pour DiagCamemberts :



Bien sûr l'implantation de ces diagrammes devra utiliser les classes Camembert et Faisceau développées au TP8.

Démonstration

Préparez la démonstration d'un programme Java qui affiche une page HTML avec un diagramme de visualisation de données à partir d'un fichier CSV. Celui-ci prend donc en paramètre :

- une chaîne spécifiant le type de diagramme souhaité : DiagBarres, DiagCamemberts ou DiagColonnes.

- un fichier CSV contenant les données à afficher, avec un format ressemblant à :

```
"Emissions_de_CO2(en_Mt)","Afrique", "Amerique", "Asie", "Europe", "Oceanie"  
"2010", 1600, 6800, 16000, 4300, 300  
"2015", 1900, 6600, 17500, 3800, 330  
"2020", 2100, 6200, 17800, 3600, 340
```

Important : Si vous n’avez pas le temps d’implémenter tous les diagrammes, concentrez vos efforts sur l’un d’entre eux afin d’avoir quelque chose à démontrer. La démonstration devra être structurée de la manière suivante :

- 1 slide de présentation de l’équipe (nom équipe, membres)
- 1 slide pour “vendre” votre produit
- 1 slide résumant ses possibilités (types de diagrammes supportés)
- une démonstration en live

le tout devant tenir en moins de 10 minutes.