

Étape 7 (NLU) [20/22]

Classe BlackBodyColor [7/8]

- 0 : Vous pourriez utiliser une liste plutôt qu'une map.
- 1 : Le flot n'est jamais fermé

Classe ObservedSky [13/14]

- 1 : Copie inutile d'une valeur immuable (unmodifiableList suffit dans les getters)
- 0 : Vous pourriez optimiser objectClosestTo en calculant la distance seulement des objets qui sont dans un carré autour du point.

Étape 8 (VSE) [17/20]

Classe SkyCanvasPainter [17/20]

Dessin des étoiles et astérismes [6/8]

DrawAsterisms:

- 1 : setLineWidth(1) manquant, les astérismes sont donc plus gros que ce qui était demandé

DrawStars:

- 1 : La constante n'est pas nommée (Angle.ofDeg(0.5) aurait du être une constante)

Dessin des planètes [4/4]

- 0 : Très bien

Dessin de la Lune et du Soleil [3/3]

DrawSun:

- 0 : La constante n'est pas nommée (Color.YELLOW.deriveColor(0,1,1,0.25) aurait du être une constante)

DrawMoon:

- 0 : Très bien

Dessin de l'horizon [4/5]

- 1 : Méthode pas très lisible avec autant de variables, vous auriez pu faire tout ça dans la boucle.

Étape 9 (DJU) [19/22]

Classe DateTimeBean [6/6]

Interface TimeAccelerator [5/5]

Type énuméré NamedTimeAccelerator [3/3]

Classe TimeAnimator [5/8]

- 3 : La mise à jour du temps de départ de l'animation à chaque appel à handle génère des bugs graphics (animation trop rapide/trop lente en jour/jour sidéral)

Étape 10 (DNO) [21/23]

Classe ObserverLocationBean [4/4]

- 0 : Interface et documentation complètes.

Classe ViewingParametersBean [3/3]

- 0 : Comme avant: interface&documentation complètes.

Classe SkyCanvasManager [14/16]

- 0 : Le commentaire de classe devrait expliqué ce que fait/ce à quoi sert la classe.
- 0 : Vous auriez pu vous contenter d'avoir les Beans passés en argument comme attributs, plutôt que de stocker chaque attribut séparément.
- 1 : La constante n'est pas statique Utiliser une map pour le déplacement du centre est élégant, mais cette map n'a pas vraiment de raison ne pas être static.
- 0 : Définir le triplet "min-max-intervalle" n'est pas obligatoire; vous auriez pu vous contenter de définir l'intervalle [mais si vous préférez faire comme cela, continuez comme cela ;)].

Constructeur:

- 0 : (Ligne 183) Évitez de déclarer des variables sans les initialiser; donnez-leur une valeur (0, 'null',...). Dans ce cas précis, vous pourriez initialiser cette variable en une ligne en utilisant l'opérateur ternaire (... ? : ...).
- 1 : Votre gestion des touches du clavier est imparfaite. Vous devriez forcer 'newAzDeg' et 'newAltDeg' dans l'intervalle attendu (en utilisant 'clip()' et 'reduce()') plutôt que ne rien faire.

Interface:

- 0 : Si vous aviez stocké les attributs sous forme de bean, vous auriez pu simplement retourner les-dits beans, simplifiant ainsi l'interface en réduisant le nombre d'accesseurs.
- 0 : Bonne mise-en-page du code (surtout au niveau de la déclaration des bindings).

Étape 11 (BTH) [18/23]

Classe Main [18/23]

Barre de contrôle [8/12]

observerPos:

- 1 : répétition de code entre la création des cases pour la longitude et la latitude, vous auriez pu mettre le code en commun dans votre méthode formater par exemple

timeManager:

- 1 : Une solution plus concise qu'un auditeur (listener) existe
- 2 : Une méthode existante aurait dû être utilisée (setZonedDateTime dans DateTimeBean pour le setOnAction de resetButton)

Panneau du ciel [2/2]

Barre d'information [4/4]

Reste de la classe [4/5]

- 1 : La classe n'est pas finale