

NOM :

PRÉNOM :

INE :

R2.01 – Conception

2023-2024

Une seule page A4 recto/verso manuscrite autorisée.

Durée : 20' – Sujet écrit sur 2 pages

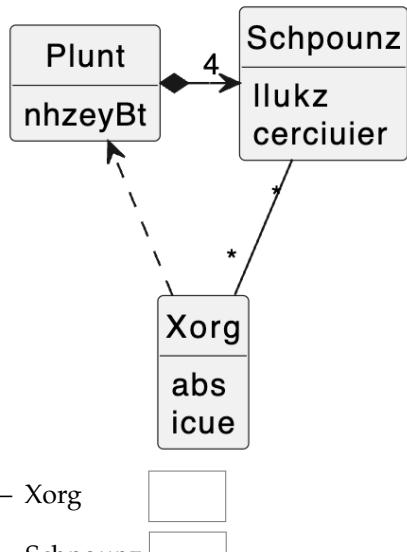
5 avril 2024

Écrivez directement sur le sujet dans les zones de réponse. Ne pas déborder de ces zones.

N'oubliez pas de mettre votre nom sur chaque page.

1 Exercice 1 – Diagramme de classe UML et code Java (≈ 3 points)

Q.1 Soit le diagramme de classe suivant. Indiquez, pour chaque classe, combien son implémentation Java aura d'attributs.



— Xorg

— Schpounz

— Plunt

Q.2 Donnez un exemple de constructeur de la classe Plunt qui respecte le diagramme de classe de la question précédente.

2 Exercice 2 – Diagramme de classe (\approx 3 points)

Q.3 Réalisez un diagramme de classe correspondant au domaine suivant (avec les attributs et les méthodes pertinentes¹⁾) :

Un artisan pêcheur souhaite développer une application permettant de gérer son entreprise de pêche. Son entreprise possède des bateaux, chacun étant identifié par son modèle, son nom, son numéro d'immatriculation et la date de mise en service. Le nombre minimum et maximum de personnes à bord, la quantité totale maximale de stock sont des données communes à tous les bateaux d'un même modèle. On ajoutera au diagramme la classe Main qui stocke dans ses attributs la liste des bateaux et des modèles.

L'artisan souhaite pouvoir obtenir facilement l'âge d'un bateau, et le tonnage maximum de sa flotte actuelle (sur l'ensemble de ses bateaux).

3 Exercice 3 – Modèles et code (\approx 4 points)

Q.4 Parmi les relations suivantes, lesquelles sont des relations possibles entre cas d'utilisation (plusieurs réponses possibles) ?

- << extend >>
 - << include >>
 - Association
 - Dépendance
 - Héritage
 - Aucune des autres réponses

Q.5 La relation d'héritage qui relie deux classes :

- Ne peut pas avoir de nom
 - Peut se lire “est un”
 - Peut se lire “a un”
 - Possède des multiplicités

Q.6 Une classe Java contient tous ses objets ?

- vrai
 - faux

Q.7 La notion de "navigation" d'une association $A \rightarrow B$ représente le fait que :

- Les objets de la classe *A* ont accès aux références d'objets de la classe *B*
 - Les objets de la classe *B* ont accès aux références d'objets de la classe *A*
 - Les objets de la classe *B* n'ont pas accès aux références d'objets de la classe *A*
 - Les objets de la classe *A* n'ont pas accès aux références d'objets de la classe *B*

1. Ne pas donner le code des méthodes.