

CONTROLE DE CONNAISSANCES STRUCTURES DE DONNÉES

Aucun document autorisé, aucune calculatrice autorisée ...

Notation : vous respecterez les notations (indices) utilisées sans quoi vous perdrez des points.

Exercice 1 : Questions de cours – 20 minutes – (≈ 6 points)

Vous répondez au maximum en 3 phrases à chaque question. Attention vous n'avez que 20 minutes. Toute mauvaise réponse entraîne la suppression de points.

1. À quoi servent les structures de données pile, file, ensemble, liste, ... ?
2. En Java, quels sont les critères qui guident le choix d'une des implémentations de l'interface List ?
3. Qu'est-ce que la généricité ?
4. À quoi sert le classpath ?
5. Qu'est-ce que le bytecode en Java ?
6. Qu'est-ce qui distingue un type primitif d'un type complexe en Java ?
7. À quoi sert le mot-clef static en Java ?
8. Comment définir une constante en Java ?
9. Il y a-t-il une erreur de lors de la compilation du code Java suivant :

```
int a = (int) true;
```

10. A quoi correspond "(T[])" ci-dessous :

```
public PileGene() {  
    sommet = -1;  
    P      = (T[])new Object[10 000];  
}
```

- A. A transtyper vers le type T[]
- B. Au paramétrage de Object.
- C. A l'allocation d'un tableau.
- D. A la création de la pile.

11. Remplacer les "..." du code ci-dessous par la bonne instruction :

```
public PileGene(PileGene<T> pile) {  
    int i;  
    sommet = pile.sommet;  
    P      = (T[])new Object[10 000];  
    for (i=0; i<=sommet; i++) {  
        ...  
    }  
}
```

- A. P[i] = pile.P[i];
- ☒ B. P[i] = this.P[i];
- C. P[i] = pile(i);
- D. P[i] = pile.sommetPile(i);

12. Remplacer les "..." du code ci-dessous par la bonne instruction :

```
public class EnsembleArray implements Ensemble {

    private int [] T;    //Tableau contenant les entiers de l'ensemble
    private int indice; //Indice de remplissage

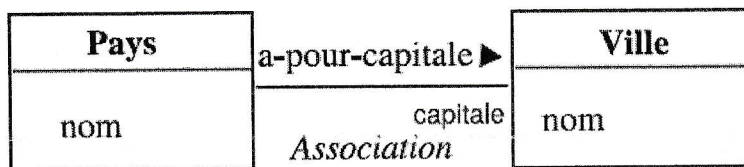
    public EnsembleArray() {
        indice = -1;
        T      = new int [10000]; //Par défaut 10000 éléments
    }

    public int nbElement() {
        ...
    }
}
```

- A. return (indice+1);
- B. return (indice);
- C. return T.lenght;
- D. return T.lenght-1;

Exercice 2 : 30 minutes - 5 points

Soit le diagramme de classe suivant :



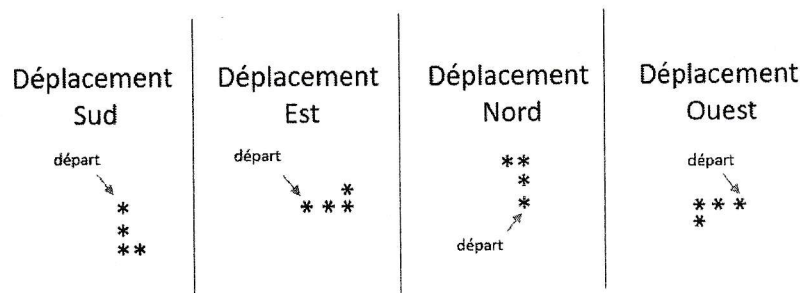
Traduisez ce diagramme en Java. Vous devez être exhaustif (constructeurs, méthodes get et set, etc) et cela doit être correct grammaticalement, syntaxiquement et algorithmiquement.

Ecrire la méthode toString() pour chacune des classes.

Ecrire une méthode main pour chacune des classes avec la création de deux instances différentes dans chaque main.

Exercice 3 : 1h10 – 9 points

Nous souhaitons définir la classe CavalierL qui ressemble à la classe Tortue vue lors du troisième TP. La classe CavalierL ressemble au concept de cavalier du jeu d'échec mais les mouvements qui nous intéressent ici sont moins nombreux. Voici les 4 mouvements possibles :



Voici l'interface de la classe CavalierL :

```
public interface CavalierL {  
  
    // codage interne suggéré : nord(0); ouest(1) ; sud(2) ; est(3);  
    // déplacement dans une direction donnée (d)  
    private void deplace(int d);  
  
    public void mouvement();  
  
    // tourne à gauche ou à droite  
    // C peut prendre la valeur « D » pour droite et « G » pour gauche  
    public void tourne(char C);  
}
```

Question 1 : ~ 2 pts

Nous souhaitons implémenter cette interface et donc créer une classe.

- a) Proposer une classe, tel que ses attributs soient les suivants :

```
private int _x, _y; // sa position courante  
private int dir;    // direction courante nord(0), ouest(1), sud(2), est(3)
```

On fait l'hypothèse que la pièce « cavalierl » regarde vers une direction (nord, ouest, sud, est) ce qui impose un déplacement « particulier ».

- b) Donner le constructeur par défaut et le constructeur par paramètres

Question 2 : ~ 3 pts

Ecrire les méthodes de la classe.

Question 4 : ~ 2 pts

Ecrire la méthode 'main' qui :

- crée un cavalierl (nommé cav1) avec le constructeur par défaut,
- crée un cavalierl (nommé cav2) avec le constructeur par paramètres,
- applique un mouvement à cav1 et à cav2.

Question 5 : ~ 2pts.

Ecrire la méthode toString (et à défaut la méthode affichage si vous ne savez pas comment écrire la méthode toString ; dans ce cas je note la question sur 1 pt).

Repère :

