

Contrôle Machine

4 Avril 2025

Consignes :

- Le contrôle dure 2h15.
- Vous avez droit à vos notes de cours, la documentation Java ainsi qu'aux TP (sujets+votre travail).
- La consultation de tout autre site est interdite.
- Le contrôle doit être fait sur une machine de l'IUT, sous OS Unix.
- La communication est évidemment interdite.
- Chaque exercice sera dans son propre dossier avec les classes correspondantes.
- Un exercice incomplet peut rapporter des points tant que le code compile et produit une partie du résultat demandé.
- Vos réponses doivent être envoyée à luc.dartois@u-pec.fr avec le sujet :
[ControleMachine] votre_Prenom votre_Nom
- Vous y joindrez une unique archive s'appelant *NomPrenom.tar* contenant les dossiers de vos exercices. Je rappelle la commande pour archiver le répertoire courant:


```
tar cvf NomPrenom.tar *
```
- Le non-respect des consignes exactes entraîne une perte de 2 points (vous aurez directement 0 si vous communiquez).

Exercice 1. Tri de caractères (4 points).

- 1) Écrire un programme prenant une chaîne de caractère en argument, et n'affichant que les caractères minuscules (a-z). Exemple :

```
java Exercice1 AbracaDaBRa  
bracaaa
```

- 2) Modifiez votre programme pour afficher tous les caractères hors minuscules dans une nouvelle ligne à la suite. Exemple :

```
java Exercice1 AbracaDaBRa  
bracaaa  
ADBR
```

Exercice 2. (6 points)

- 1) Créez une classe *DevinerEntier* qui contient un attribut privé entier, et deux constructeurs : un prenant un entier, et un ne prenant pas d'argument et assignant une variable aléatoire entre 0 et 100. Vous pourrez vous aider de la classe [Random](#) du package *util* pour générer l'entier aléatoire.
- 2) Ajoutez une méthode *estPlusPetitQue* qui prend en entrée un entier et le compare à l'attribut de votre classe. Elle doit renvoyer :
 1. -1 si l'entier de la classe est plus petit que l'argument.
 2. 0 s'ils sont égaux
 3. 1 sinon.
- 3) Écrivez un programme qui génère un *DevinerEntier*, puis cherche sa valeur et l'affiche. Le programme devra afficher chaque essai de façon claire.

Exercice 3. Feu Tricolore (7 points).

- 1) Créer une classe *Cercle* héritant de *JComponent* affichant un cercle. Le composant contiendra un attribut [Color](#). Un objet *Cercle* pourra être allumé (i.e. colorié de sa couleur) ou éteint (colorié en noir). La classe *Cercle* pourra contenir toute méthode ou attribut jugé utile.
- 2) Écrire une classe *FeuTricolore* contenant et affichant 3 composants *Cercle* respectivement de couleurs Rouge, Orange et Vert, et les affichant les uns en dessous des autres.
- 3) Écrire une classe de Test pour votre classe *FeuTricolore*.
- 4) Faites en sorte que lorsque l'on clique sur un composant, il s'allume et les deux autres Cercles s'éteignent.

Exercice 4. (5 points).

Voici l'interface *TirageAleatoire* :

```
public interface TirageAleatoire{
    public int nextInt();
}
```

- 1) Créer des classes *Piece* et *DeSix* qui réalisent l'interface *TirageAleatoire*. *Piece* renvoie un entier entre 0 et 1, alors que *DeSix* renvoie un entier entre 1 et 6. Vous pourrez vous appuyer sur la classe [Random](#).

La méthode suivante permet de faire des tirages aléatoires et d'en sommer le résultat :

```
public static int sommeIterative(TirageAleatoire ta,int n){
    int result=0;
    for(int i=0;i<n;i++){
        result+=ta.nextInt();
    }
    return result;
}
```

- 2) Utilisez cette méthode pour obtenir et afficher le résultat de 1000 lancers de pièces.
- 3) De même, simulez 10000 lancers de Dé à 6 faces, et affichez la moyenne des valeurs obtenues.