TP 11 Classes et objets

Informations techniques PC Suse :

(a) Pour démarrer une session : utilisateur **licencep** et mot de passe **7002n***. Vous trouverez :
— un gestionnaire de fichiers en haut à gauche placé dans le dossier personnel HOME
— une `icône lézard`  en haut à droite pour accéder au menu.

(b) Pour démarrer *Processing* : clic sur `l'icône lézard` en haut à droit → Dévelopemment → Processing.

(c) Pour démarrer une *console* : clic sur `l'icône lézard` en haut à droit → Terminal → Konsole.

(d) Pour ouvrir un gestionaire/navigateur de fichiers : clic sur `l'icône lézard` → Utilitaires → Dolphin.

(e) Pour lancer une commande : Alt + Space

(f) Pour modifier un fichier, clic droit sur le fichier → Ouvrir avec → Kate (ou autre éditeur de votre choix).

1 Commandes Linux et un petit programme Java

Exercice 1 Ouvrir un terminal, taper la commande suivante pour s'assurer d'être bien placé dans le dossier personnel

```
cd
```

Exercice 2 Taper la commande suivante pour trouver tous les fichiers `.java` (c.à.d, d'extension `java`) dans le dossier personnel (même ceux d'autres auditeurs).

```
find . -name "*java"
```

Exercice 3 Taper une commande `mkdir` (*make diretory*) pour créer un dossier appelé `tp11`. Taper une commande `cd` pour se placer dans le dossier que vous venez de créer.

Exercice 4 Taper la commande ci-dessous pour télécharger le fichier source `Toto6.java`.

```
wget  cedric.cnam.fr/~porumbed/Toto6.java
```

Compiler et exécuter `Toto6.java` à l'aide des instructions fournies au TP6, TP7, TP8, TP9 ou TP10.

Exercice 5 Chercher le fichier `Toto6.java` dans le gestionnaire de fichiers. Clic droit sur `Toto6.java` et choisir `Ouvrir avec` → `kate`. Vous pouvez faire la même chose grâce à la commande

```
kate  Toto6.java&
```

Grâce au `&`, la commande ne bloque pas le terminal

Ajouter une boucle `for` pour faire le programme calculer et **afficher la somme** des valeurs du tableau.

2 Classes et objets

Exercice 1 Définir une classe `Compte` pour faire fonctionner le code ci-après.

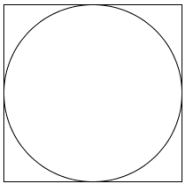
1. Le constructeur doit avoir un seul argument qui initialise un champ/attribut `titulaire`.
2. La classe compte devrait permettre le dépôt et le retrait d'une somme d'argent passée comme argument aux méthodes `depot(...)` ou `retrait`.
   — Vous allez avoir besoin d'un attribut `solde` pour stocker le solde de chaque objet.
3. N'oubliez pas de définir la méthode `toString` qui permet de convertir un objet de type `Compte` en chaîne de caractères. Rappel : la méthode `toString()` est appelé automatiquement à chaque appel comme `println(c2)`.

```
.....
void setup(){
    Compte c1 = new Compte("Ada");
    Compte c2 = new Compte("Nathalie");
    c1.depot(20);   //Ada possède 20euros
    c1.retrait(10);//Ada possède 10euros
    println(c2);    //Afficher "Ada: 10"
}
```

Exercice 2 Le code ci-après (page suivante) utilise un tableau d'objets de type `Compte`. On observe qu'on donne à chaque compte une somme d'argent aléatoire à la ligne 13.
Consigne : ajouter une fonction `getSolde()` à la classe `Compte` qui renvoie le solde du compte en question. Afficher le nom de la personne la plus fortuné (avec un solde maximum).

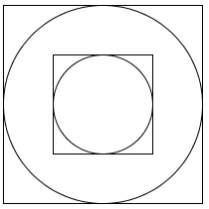
```
class Compte{
  //copier le code de l'exo précédent
  //ajouter la fonction getSolde()
  //pour renvoyer le solde
}
void setup(){
  Compte[] c = new Compte[4];
  c[0] = new Compte("Ada");
  c[1] = new Compte("Nathalie");
  c[2] = new Compte("Daniel");
  c[3] = new Compte("Alice");
  for(int i=0;i<4;i++)
      c[i].depot((int)random(200));
  for(int i=0;i<4;i++)//on affiche les
    println(c[i].getSolde());//soldes
}
```

On considère une figure géométrique constitué d'un cercle inscrit dans un carré, comme dans la figure à droite. Une telle figure est complètement déterminé par le centre (cx, cy) et par le rayon r ; le coté du carré mesure $2r$.



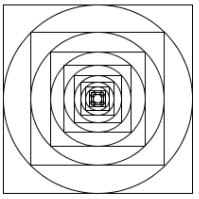
Exercice 3 Définir une classe `Figure` permettant de manipuler de telles figures géométriques. Définir un constructeur prenant en paramètre 3 valeurs `float` qui indiquent le centre (deux coordonnées) et le rayon.

Exercice 4 Continuer la classe `Figure` et ajouter une méthode `dessiner()` qui permet de tracer la figure sur la toile. Construire dans `setup()` deux objets de type `Figure` de même centre pour les tracer. Ce code devrait tracer la figure à droite.



```
class Figure{
...
}
void setup(){
  size(500,500);
  background(255,255,255);
  Figure f = new Figure(200,200,100);
  f.dessiner();
  Figure fPetit = new Figure(200,200,50);
  fPetit.dessiner();
}
```

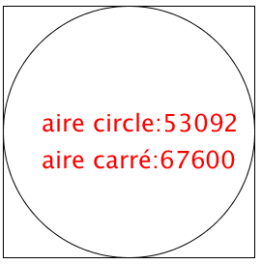
Exercice 5 Continuer le programme pour le faire utiliser un tableau de 10 objets de type `Figure`. Le programme ci-dessus devrait réaliser cela. Remarquer la construction du tableau de 10 cases, suivie de la construction de chaque `Figure`.



```
void setup(){
  size(500,500);
  background(255,255,255);
  Figure[] f= new Figure[10];
  float rayon = 100;
  for(int i=0;i<10;i++){
      f[i] = new Figure(200,200,rayon);
      rayon = rayon/sqrt(2);
  }
  for(int i=0;i<10;i++)
      f[i].dessiner();
}
```

Exercice 6 Écrire une méthode `surfaceCercle()` qui calcule et renvoie la surface (aire) du cercle ($PI \cdot r^2$). Écrire une méthode `surfaceCarre()` qui renvoie la surface du carré. Vérifier si le calcul est correct, par exemple, vérifier que la surface du cercle est plus petite que celle du carré. Modifier la méthode `dessiner()` pour la faire tracer en rouge les deux aires (converties en `int`). Le programme ci-après devrait tracer la figure ci-dessous.

```
void setup(){
  size(500,500);
  Figure f;
  f=new Figure(200,200,130);
  f.dessiner();
}
```



Exercice 7 Écrire un méthode `deplacer(float x, float y)` qui permet de déplacer le centre de la figure aux coordonnées `(x,y)`. Écrire une méthode `contientPoint(float a, float b)` qui teste si le point donné en paramètre (a, b) est à l'intérieur du carré.

Exercice 8 Soit une surface de taille 1000×800. Mettre au centre une grande figure de rayon 400. Construire un tableau de $n = 20$ figures plus petites, toutes de rayon 40. Appeler `deplacer(random(1000),random(1000))` sur chaque figure pour la faire déplacer. Afficher toutes les figures avec le centre à l'intérieur du carré de la première grande figure.

Exercice 9 Finir le dernier exercice du TP précédent sur le jeu `casse-briques`.

Exercice 10 Faire l'animation 3D décrite à `cedric.cnam.fr/~porumbed/vari1/anim3d/`
→ Montrez moi si vous arrivez à faire cet exercice, cela pourrait vous apporter des points bonus.