

TP 5 VARI 1

cedric.cnam.fr/~porumbed/vari1/

Informations techniques PC Suse :

- (a) Pour démarrer une session : utilisateur **licencep** et mot de passe **7002n****.
- (b) Pour démarrer *Processing* : clic sur la tête de coméléon en haut à droite → Développement → Processing.
- (c) La page VARI1 : cedric.cnam.fr/~porumbed/vari1/
- (d) Pour ouvrir un gestionnaire/navigateur de fichiers : clic sur la tête de caméléon → Système → Dolphin.

Avant-propos Vous êtes encouragés à passer en mode actif, c.à.d., utiliser les méthodes `setup()` et `draw()`, voir les corrections vidéo du TP 4 ou le cours 5.

Exercice 1 Tracer un triangle non-rempli avec ses trois sommets situés au pixels : (0,0), (100, 100), (20, 70). **Indication** : L'appel `triangle(x1, y1, x2, y2, x3, y3)` permet de tracer un triangle déterminé par les points/sommets (x1,y1), (x2,y2), (x3,y3).

Exercice 2 Réaliser le dessin précédent grâce à trois appels à `line(...)`.

Exercice 3 Utiliser six appels `triangle(...)` pour réaliser le dessin à droite, en deux couleurs (rouge et vert).

Exercice 4 Utiliser deux appels `quad(...)` pour réaliser le dessin à droite, de couleur verte. Indication :

- L'appel `quad(x1, y1, x2, y2, x3, y3, x4, y4)` permet de tracer un quadrilatère déterminé par les points/sommets (x1,y1), (x2,y2),...

Exercice 5 Réaliser le dessin de l'exercice précédent sans fait appel `quad`. Au lieu d'utiliser `quad`, vous pouvez faire appel à `beginShape()`, `vertex(...)` et `endShape()`, comme à l'exercice 6 du TP précédent.

Exercice 6 Écrire un programme *processing* qui permet d'initialiser une variable `salaire` de type `float` à une valeur au hasard entre 300 euros et 1000000 euros. Afficher un des messages suivants :

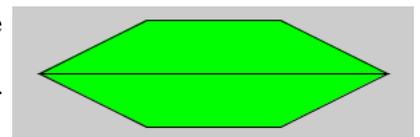
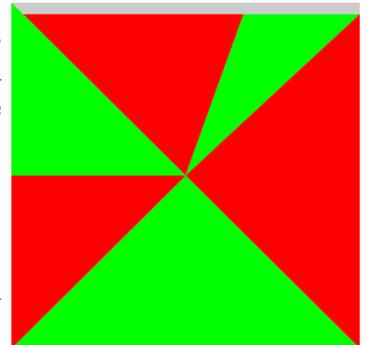
- Si le salaire est inférieur ou égal à 10000, afficher « Vous n'êtes pas imposable »
- Si le salaire est supérieur à 10000 mais inférieur ou égal à 100000, afficher « Vous payez un impôt de 10% ».
- Si le salaire est supérieur à 100000, afficher « Vous êtes footballeur ? Vous payez un impôt sur la fortune de 30% »

Finalement, afficher le salaire **net**, après la déduction de l'impôt.

Exercice 7 Écrire un programme qui permet de tracer un hexagone de couleur rouge, sans contour ; vous pouvez utiliser deux appels à `quad(...)`. À l'intérieur de l'hexagone, tracer un cercle de couleur bleu et écrire « Salut Toto » à l'intérieur du cercle.

Exercice 8 Corriger les erreurs de compilations du programme ci-après et exécuter ce programme.

```
1 int zEntier=0, x = 4;
2 float z = random(3)
3 String str = "x="+x; //str est une chaine de caractères
4 println(str); //on affiche cette chaine qui comporte la valeur de x
5 println(str.length()); //on affiche le nombre de caractères, toujours 3, pourquoi?
6 int zEntier = (int)z //le '(int)' devant 'z' fait une conversion vers int
7 if(zEntier===2)
8     println("z vaut deux");
```



Exercice 9 Déclarer une variable entière x (de type `int`) et une variable réelle y (de type `float`). Initialiser $x = 2$ et $y = 0.5$. On veut stocker dans une variable z la somme $z = x + y$. Déclarer la variable z (quel type choisir ?), faire l'affectation $z = x + y$ et finalement afficher z à l'aide de la fonction `println(...)`.

Exercice 10

Modifier le programme ci-après pour le faire rebondir la balle lorsqu'elle touche le bord à droite. N'hésitez pas à regarder le début de la vidéo 5 pour vous familiariser avec les méthodes `setup()` et `draw()`.

```
1 int x; //variable globale visible dans chaque méthode
2 void setup() {
3     size(600,600);
4     x = 0;
5 }
6 void draw() {
7     background(100,100,100); //effacer la toile
8     x = x + 1;
9     ellipse(x,300,10,10);
10 }
```

Exercice 11 Modifier le programme précédent pour faire démarrer la balle sur en mouvement en diagonale, c.à.d., on passe d'un pixel (x, y) au pixel $(x + 1, y + 1)$, ensuite à $(x + 2, y + 2)$, $(x + 3, y + 3)$, etc. La balle doit démarrer à une position aléatoire sur la toile. Lorsque la balle touche un côté/bord de la toile elle doit rebondir. Par exemple, si la position $(x + 3, y + 3)$ touche le côté droit de la toile, alors la balle doit rebondir vers la gauche et continuer sur la trajectoire : $(x + 3, y + 3) \rightarrow (x + 2, y + 4) \rightarrow (x + 1, y + 5)$, etc.

Exercice 12 Ouvrir un terminal : tête caméléon en haut à droit \rightarrow Terminal \rightarrow Konsole. Taper la commande suivante pour télécharger le fichier `cercles.png` et utiliser le gestionnaire de fichiers, voir point (c) tout au début du document pour l'ouvrir :

```
wget cedric.cnam.fr/~porumbed/vari1/cercles.png
```

Réaliser ce dessin sous `processing`. Vous allez avoir besoin de 5 couleurs olympiques à obtenir avec les appels : `stroke(21,195,245)`, `stroke(0,0,0)`, `stroke(184,20,55)`, `stroke(245,184,8)`, `stroke(49,255,0)`.

Exercice 13 (bonus) Continuer l'exercice 10 pour ajouter la fonctionnalité suivante. Placer au centre de la toile un cercle rouge de rayon 100 légèrement transparent. La balle peut traverser ce cercle comme si ce cercle n'était pas un obstacle. Mais le contact avec le cercle génère une deuxième balle qui doit partir en sens inverse par rapport à la première. Regardez la vidéo à l'URL suivante :

<http://cedric.cnam.fr/~porumbed/exo13.gif>

Exercice 14 (bonus) Continuer l'exercice 12 pour réaliser le logo olympique disponible à l'adresse :

```
wget cedric.cnam.fr/~porumbed/vari1/drapeauolympique.png
```

Ce dessin a été réalisé avec `Processing` grâce à plusieurs appels `arc(...)`. Certaines sections des cercles en haut doivent effacer les cercles en bas.