
TP2 : POO en JS et en TS

Ayoub KARINE (ayoub.karine@isen-ouest.yncrea.fr)

Objectif :

L'objectif de ce TP est de manipuler la programmation orientée objet en JavaScript et en TypeScript.

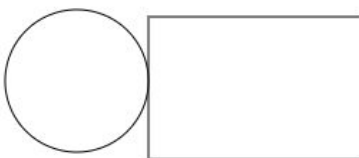
Partie 1 : Implémentation en JS :

Il est demandé de générer via le code JS suivant (main.js) :

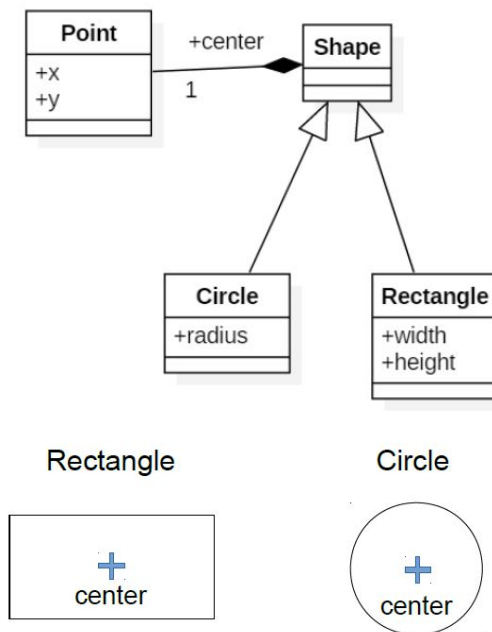
```
function drawShapes(shapes, context){
    for(let shape of shapes){
        shape.draw(context);
    }
}

function main(){
    let circle = new Circle(50);
    circle.setCenter(new Point(100,75));
    let rectangle = new Rectangle(150, 100);
    rectangle.setCenter(new Point(150,30));
    let shapes = [circle, rectangle];
    let canvas = document.getElementById("drawings");
    let context = canvas.getContext("2d");
    drawShapes(shapes, context);
}
main();
```

le dessin ci-dessous :



1. Implémenter le diagramme UML suivant en JavaScript (chaque classe doit être dans un fichier distinct)



2. Dans chaque forme, implémentez une méthode `draw(context)` permettant de l'afficher dans le contexte 2D `context` d'un canvas. Ci-dessous des bouts de code qui peuvent être utilisés dans les deux formes.

Liens utiles :

https://www.w3schools.com/tags/canvas_rect.asp

https://www.w3schools.com/tags/canvas_arc.asp

```
// Rectangle
context.beginPath();
context.rect(..., ..., ..., ...);
context.stroke();
```

```
//Circle
ctx.beginPath();
ctx.arc(..., ..., ..., 0, 2 * Math.PI);
ctx.stroke();
```

Partie 2 : Implémentation en TS

Installation du compilateur TypeScript (tsc)

1. Télécharger et installer Node.js dans le terminal de VScode. Cela va permettre d'installer Node.js Package Manager (npm)
2. Cliquer sur Terminal -> New Terminal
3. Installer TypeScript en tapant sur le terminal
4. Tester l'installation de TypeScript en tapant sur le terminal

```
npm install -g typescript
```

```
tsc --version
```

5. Créer un dossier puis l'ouvrir sur VS code
6. Créer dans ce dossier un fichier test.ts dont le contenu est

```
let message: string = 'Hello World';  
console.log(message);
```

7. Pour compiler test.ts,

- a. taper dans le terminal :

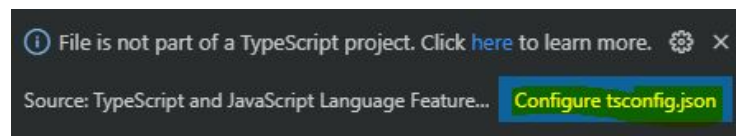
```
tsc test.ts
```

Cela va créer un fichier test.js.

- b. Comparer les fichiers test.ts et test.js

8. Pour déboguer un fichier .ts

- a. cliquer sur view>Command Palette, taper typescript puis cliquer sur TypeScript: Go To Project Configuration ensuite cliquer sur Configure tsconfig.json



- b. dans le terminal, taper tsc. Un fichier test.js.map va être ajouté
- c. ajouter un fichier test.html qui appelle le fichier test.js puis lancer le débogage sur VS code en marquant un breakpoint dans la deuxième ligne du fichier test.ts

Programmation orientée objet en TypeScript (tsc)

1. Refaire le code JS proposé dans la première partie de ce TP en utilisant le langage TS. Pour ce faire, il est demandé de ne pas utiliser les même noms de classe que la partie 1. Dans ce cas de figure le code de test est :

```
function drawShapes1(shapes, context){
    for(let shape of shapes){
        shape.draw1(context);
    }
}

function main1(){
    let circle = new Circle1(50);
    circle.setCenter1(new Point1(100,75));
    let rectangle = new Rectangle1(150, 100);
    rectangle.setCenter1(new Point1(150,30));
    let shapes = [circle, rectangle];
    const canvas: any = document.getElementById("drawings");
    const context: CanvasRenderingContext2D= canvas.getContext("2d");
    drawShapes1(shapes, context);
}
main1();
```