

# Inheritance



# Inheritance

# Inheritance

- TypeScript supports the object-oriented concept of inheritance

# Inheritance

- TypeScript supports the object-oriented concept of inheritance
  - Define common properties and methods in the superclass

# Inheritance

- TypeScript supports the object-oriented concept of inheritance
  - Define common properties and methods in the superclass
  - Subclasses can extend superclasses and add properties and methods

# Inheritance

- TypeScript supports the object-oriented concept of inheritance
  - Define common properties and methods in the superclass
  - Subclasses can extend superclasses and add properties and methods
  - Support for abstract classes and overriding

# Inheritance

- TypeScript supports the object-oriented concept of inheritance
  - Define common properties and methods in the superclass
  - Subclasses can extend superclasses and add properties and methods
  - Support for abstract classes and overriding

# Inheritance

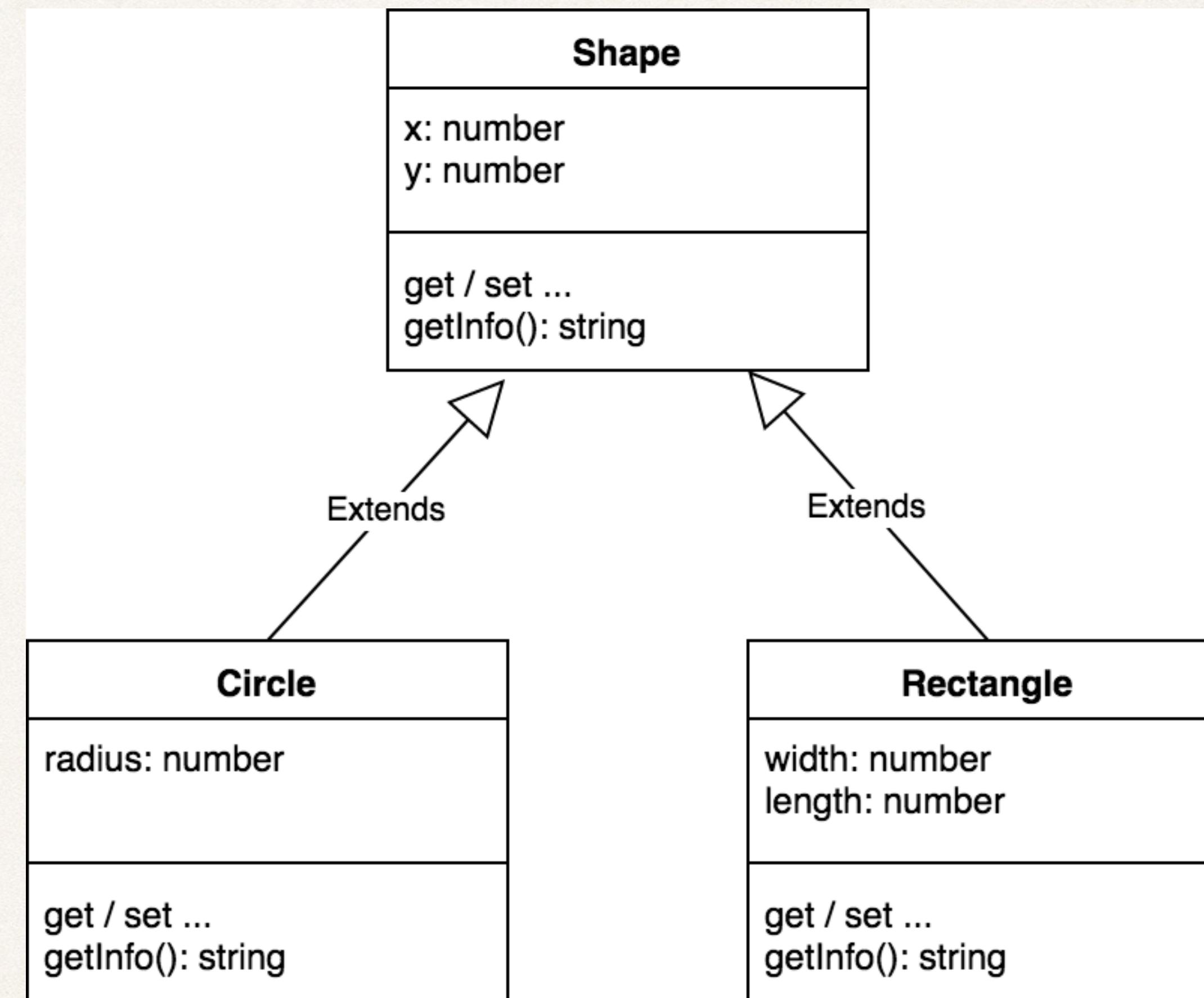
- TypeScript supports the object-oriented concept of inheritance
  - Define common properties and methods in the superclass
  - Subclasses can extend superclasses and add properties and methods
  - Support for abstract classes and overriding
- TypeScript only supports single inheritance

# Inheritance

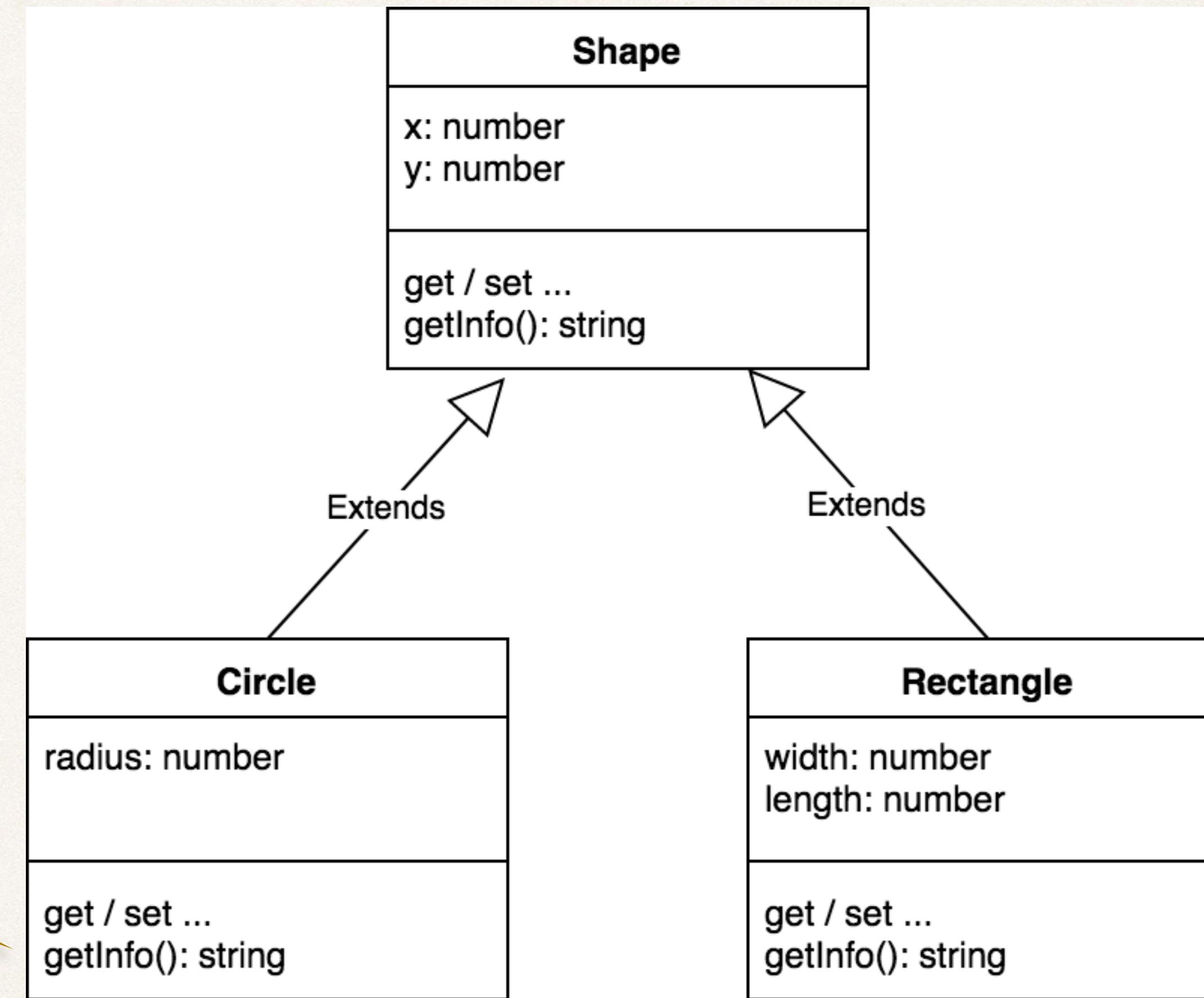
- TypeScript supports the object-oriented concept of inheritance
  - Define common properties and methods in the superclass
  - Subclasses can extend superclasses and add properties and methods
  - Support for abstract classes and overriding
- TypeScript only supports single inheritance
  - However, you can implement multiple interfaces

# Inheritance Example

# Inheritance Example

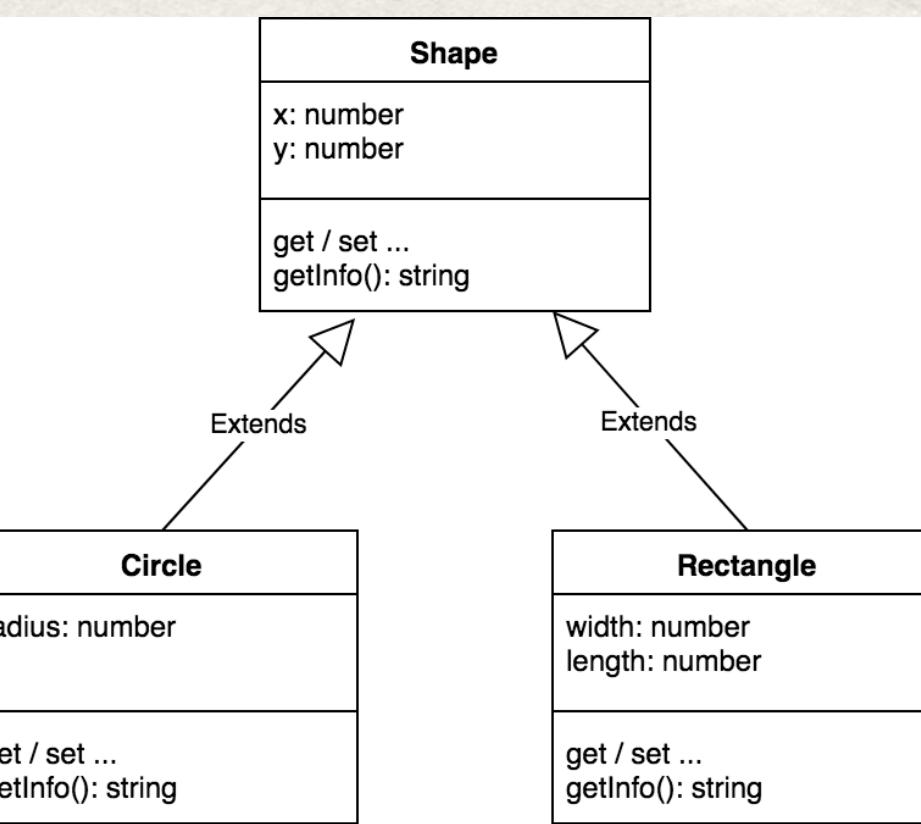


# Inheritance Example



Can override the  
getInfo() method

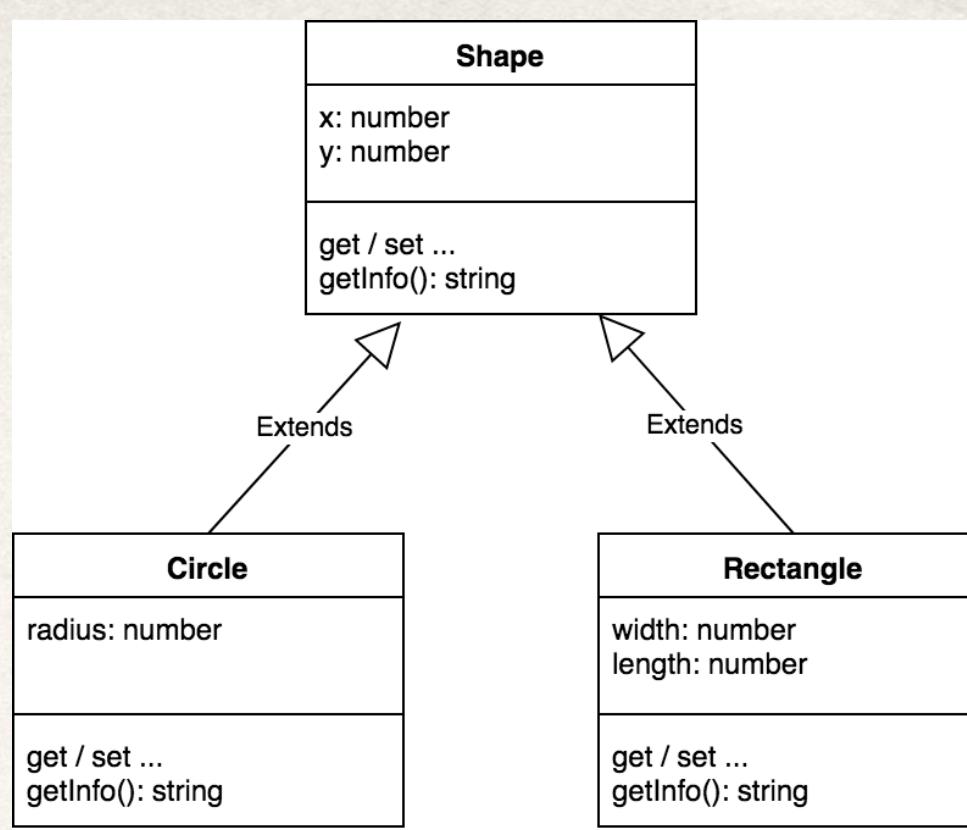
# Inheritance Example



# Inheritance Example

File: Shape.ts

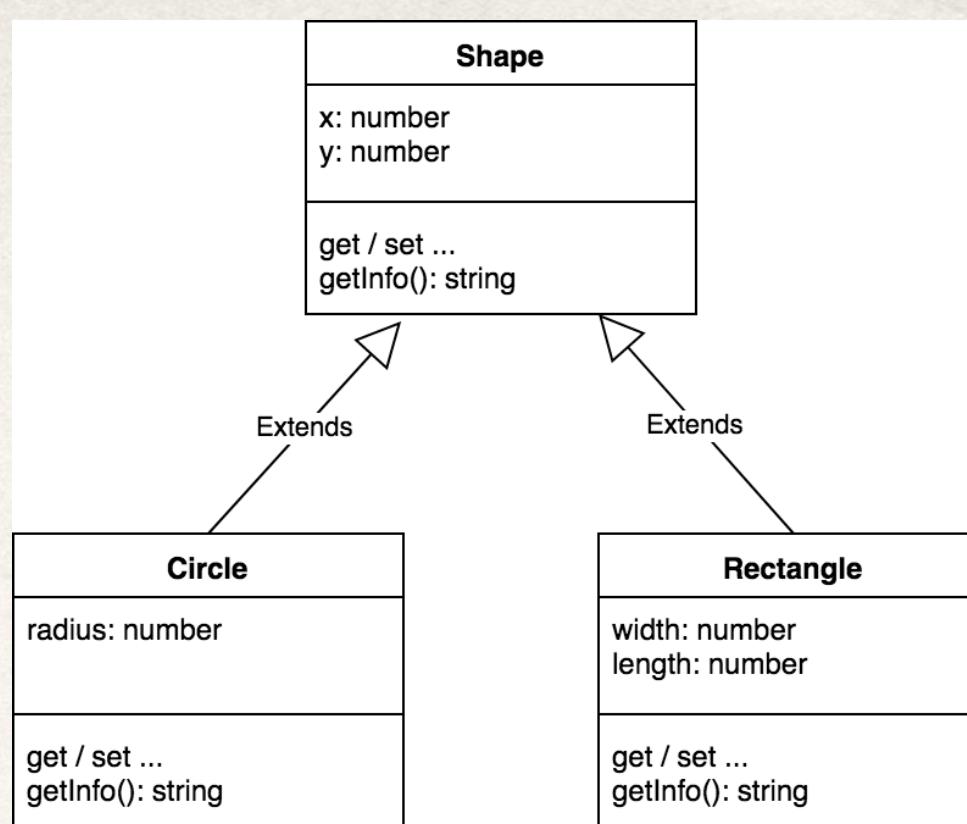
```
export class Shape {
```



# Inheritance Example

File: Shape.ts

```
export class Shape {  
  
    constructor(private _x: number, private _y: number) {  
    }  
}
```

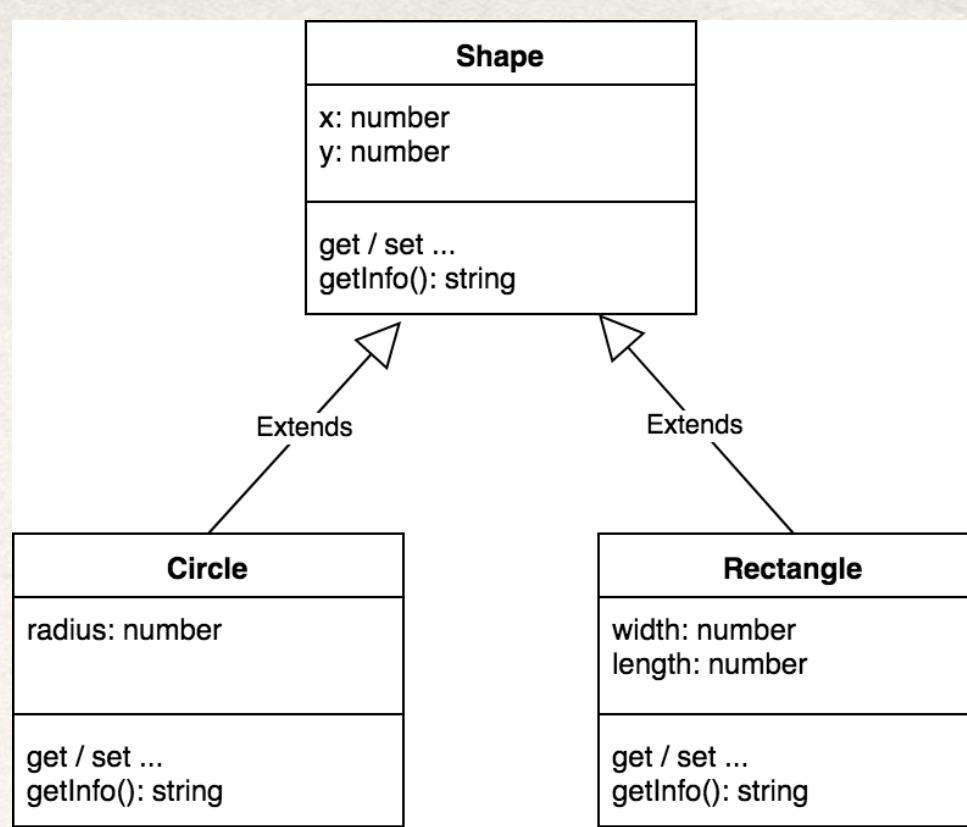


# Inheritance Example

File: Shape.ts

```
export class Shape {  
  
    constructor(private _x: number, private _y: number) {  
    }  
}
```

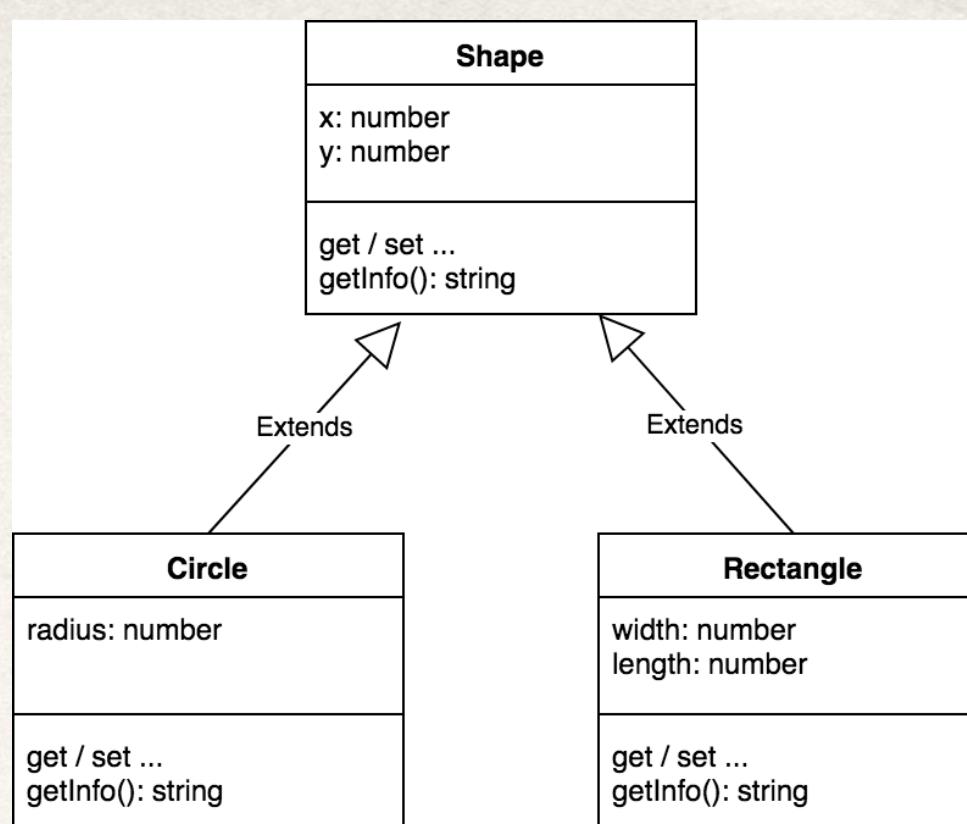
Parameter  
Properties



# Inheritance Example

File: Shape.ts

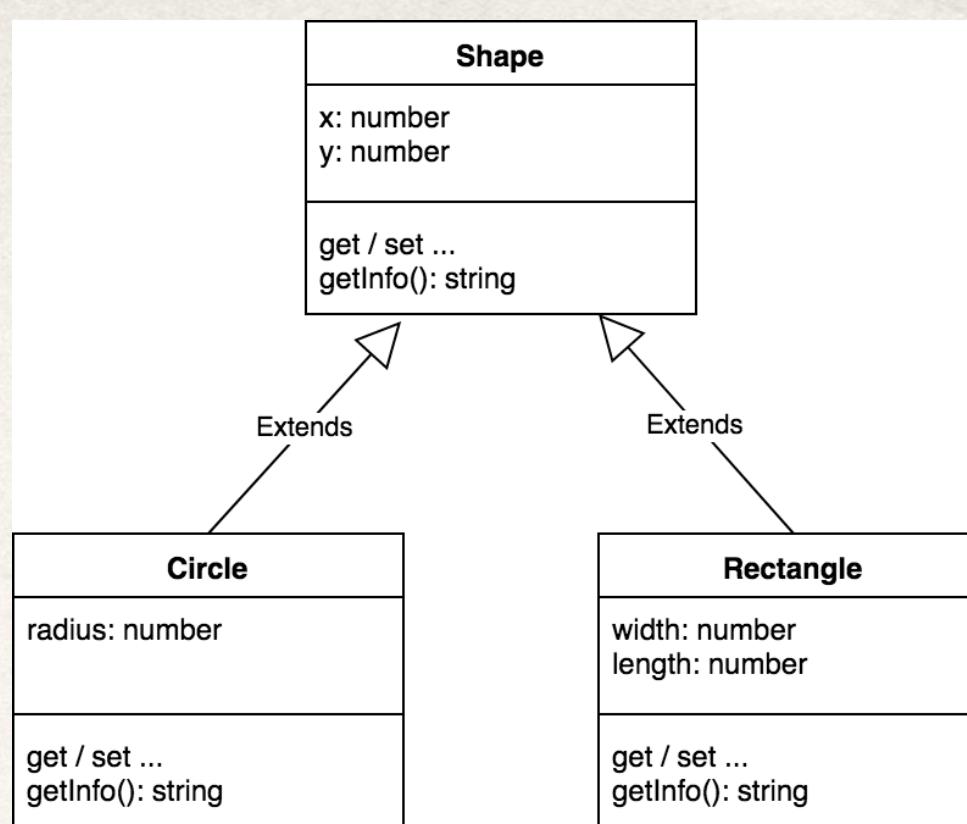
```
export class Shape {  
  
    constructor(private _x: number, private _y: number) {  
  
    }  
  
    // get/set accessors ...  
}
```



# Inheritance Example

File: Shape.ts

```
export class Shape {  
  
    constructor(private _x: number, private _y: number) {  
    }  
  
    // get/set accessors ...  
  
    getInfo(): string {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```



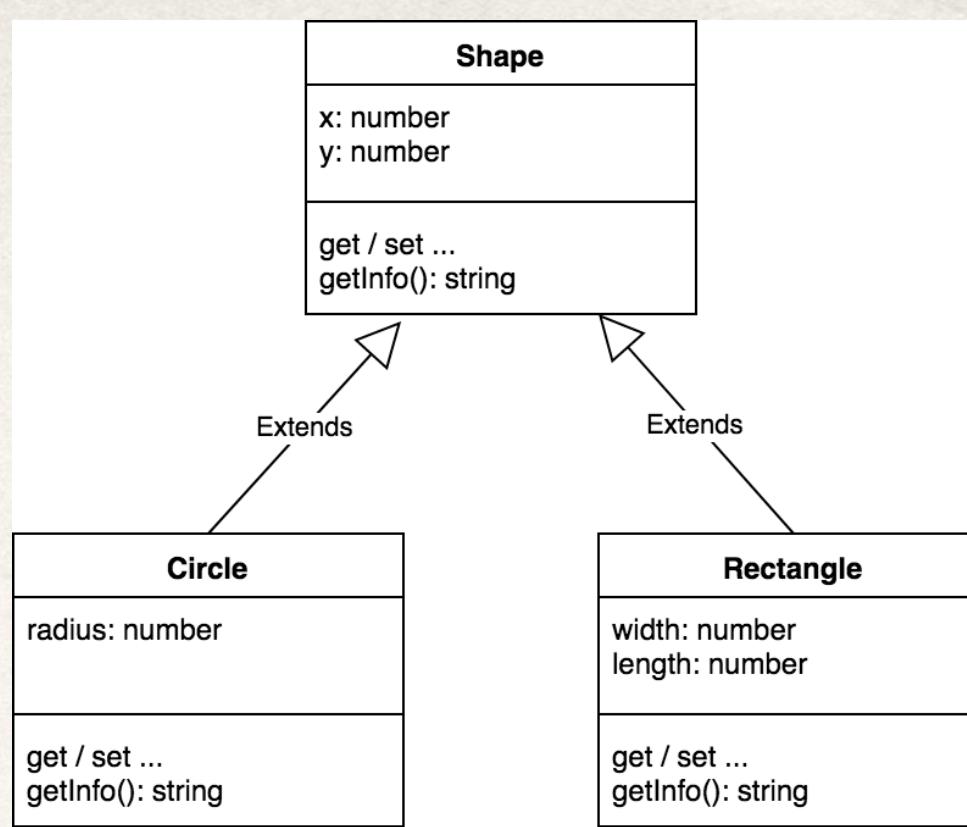
# Inheritance Example

File: Shape.ts

```
export class Shape {  
  
    constructor(private _x: number, private _y: number) {  
    }  
  
    // get/set accessors ...  
  
    getInfo(): string {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

File: Circle.ts

```
import { Shape } from './Shape';
```



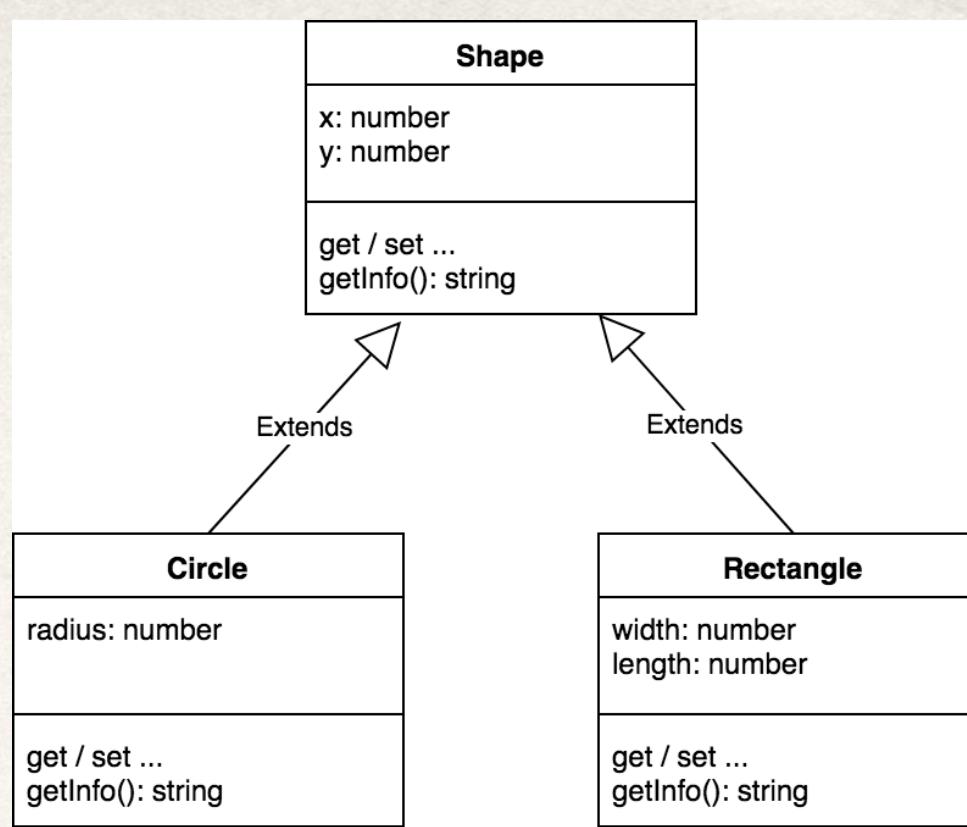
# Inheritance Example

File: Shape.ts

```
export class Shape {  
  
    constructor(private _x: number, private _y: number) {  
    }  
  
    // get/set accessors ...  
  
    getInfo(): string {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

File: Circle.ts

```
import { Shape } from './Shape';  
  
export class Circle extends Shape {
```



# Inheritance Example

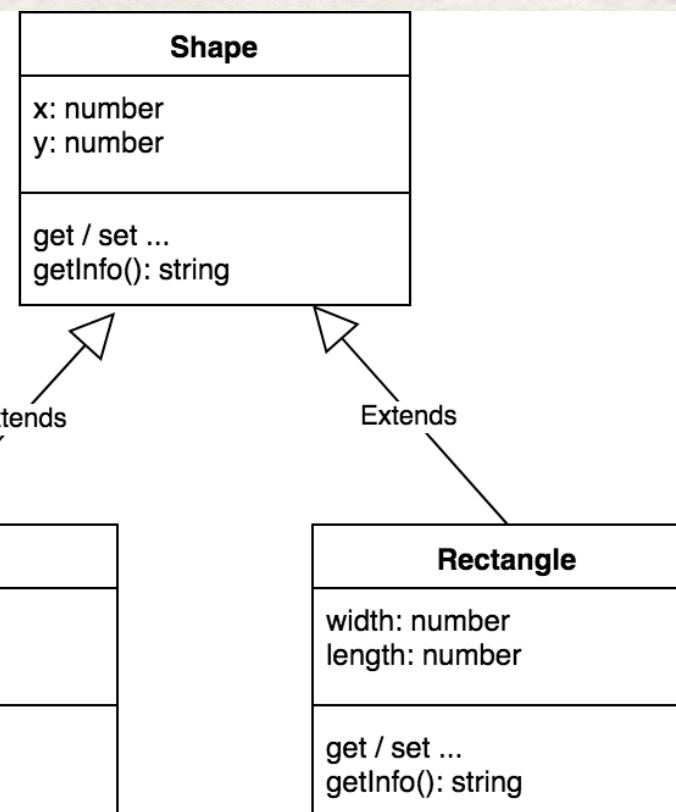
File: Shape.ts

```
export class Shape {  
  
    constructor(private _x: number, private _y: number) {  
    }  
  
    // get/set accessors ...  
  
    getInfo(): string {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

File: Circle.ts

```
import { Shape } from './Shape';  
  
export class Circle extends Shape {
```

Inheritance



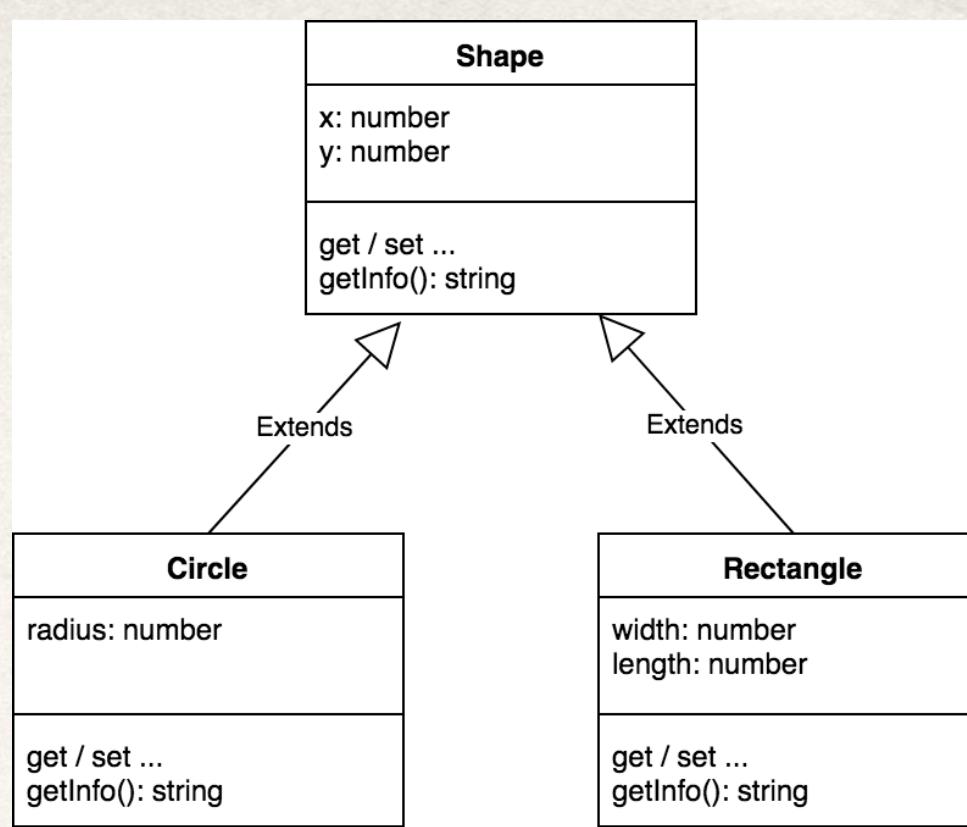
# Inheritance Example

File: Shape.ts

```
export class Shape {  
  
    constructor(private _x: number, private _y: number) {  
    }  
  
    // get/set accessors ...  
  
    getInfo(): string {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

File: Circle.ts

```
import { Shape } from './Shape';  
  
export class Circle extends Shape {  
  
    constructor(theX: number, theY: number,  
              private _radius: number) {  
        super(theX, theY);  
    }  
}
```



# Inheritance Example

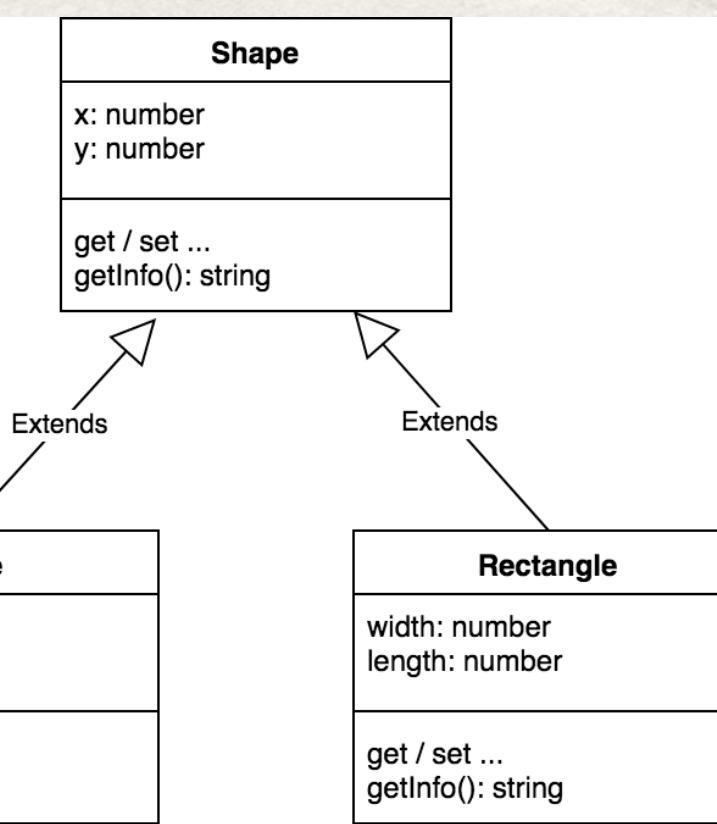
File: Shape.ts

```
export class Shape {  
  
    constructor(private _x: number, private _y: number) {  
    }  
  
    // get/set accessors ...  
  
    getInfo(): string {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

File: Circle.ts

```
import { Shape } from './Shape';  
  
export class Circle extends Shape {  
  
    constructor(theX: number, theY: number,  
              private _radius: number) {  
        super(theX, theY);  
    }  
}
```

Regular parameters  
theX and theY



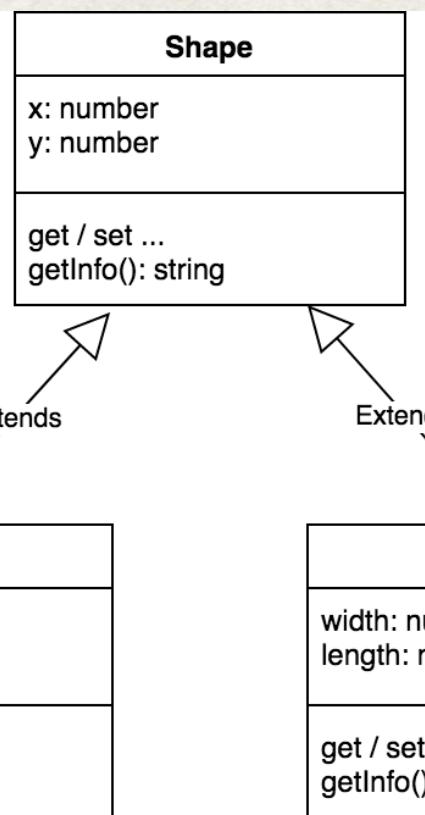
# Inheritance Example

File: Shape.ts

```
export class Shape {  
  
    constructor(private _x: number, private _y: number) {  
    }  
  
    // get/set accessors ...  
  
    getInfo(): string {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

File: Circle.ts

```
import { Shape } from './Shape';  
  
export class Circle extends Shape {  
  
    constructor(theX: number, theY: number,  
              private _radius: number) {  
        super(theX, theY);  
    }  
}
```



Regular parameters  
theX and theY

Parameter Property  
\_radius

# Inheritance Example

File: Shape.ts

```
export class Shape {  
  
    constructor(private _x: number, private _y: number) {  
    }  
  
    // get/set accessors ...  
  
    getInfo(): string {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

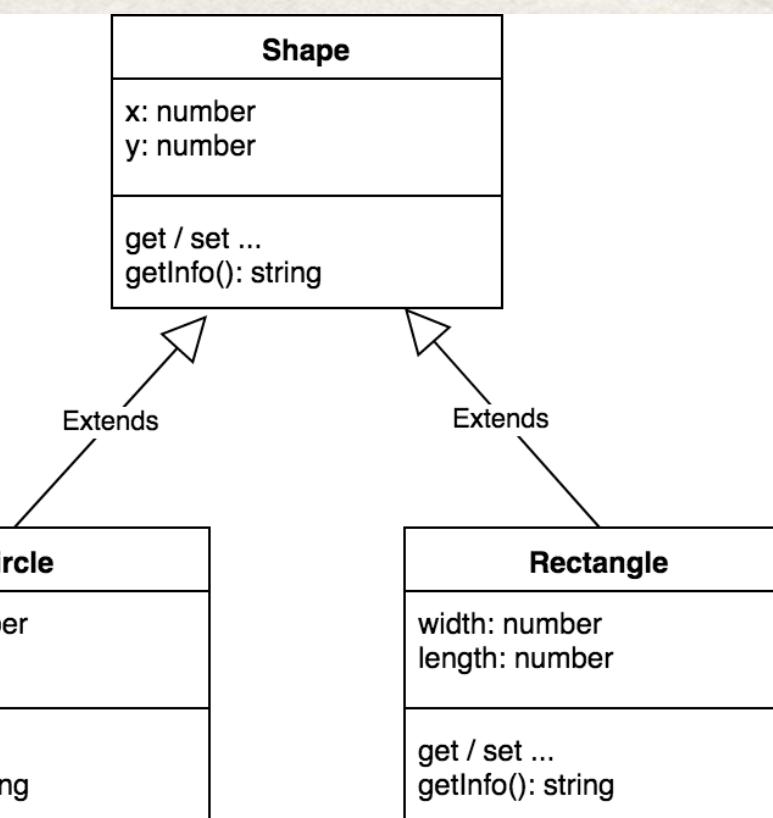
File: Circle.ts

```
import { Shape } from './Shape';  
  
export class Circle extends Shape {  
  
    constructor(theX: number, theY: number,  
              private _radius: number) {  
        super(theX, theY);  
    }  
}
```

Call superclass  
constructor

Regular parameters  
theX and theY

Parameter Property  
\_radius



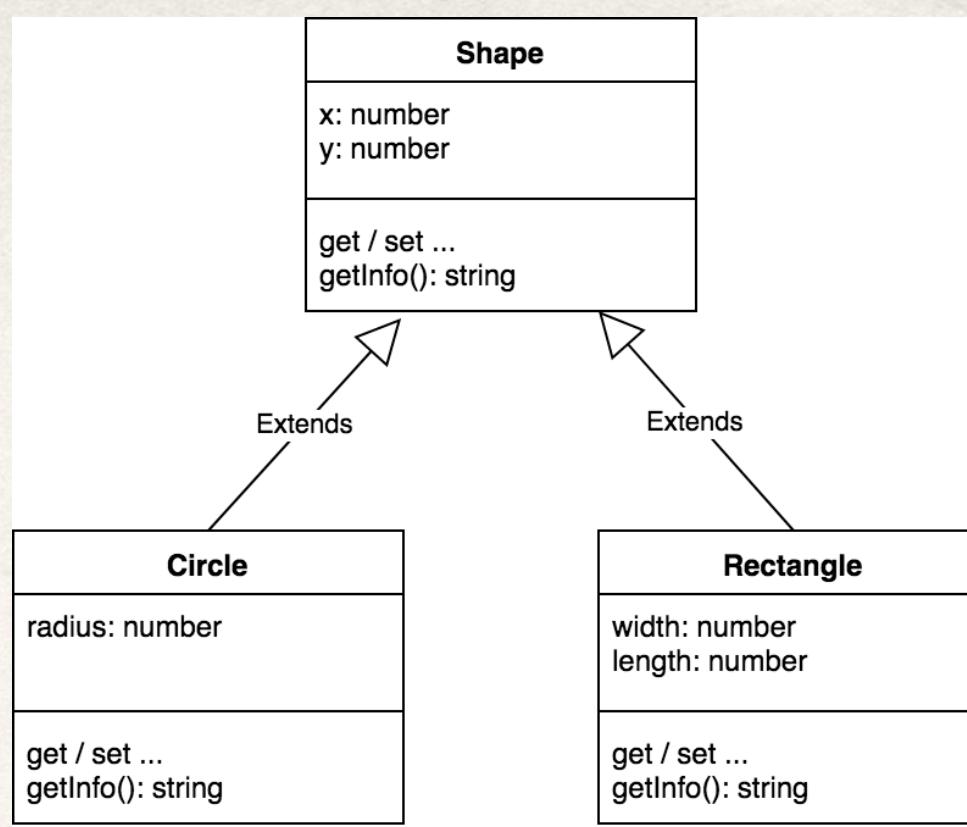
# Inheritance Example

File: Shape.ts

```
export class Shape {  
  
    constructor(private _x: number, private _y: number) {  
    }  
  
    // get/set accessors ...  
  
    getInfo(): string {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

File: Circle.ts

```
import { Shape } from './Shape';  
  
export class Circle extends Shape {  
  
    constructor(theX: number, theY: number,  
              private _radius: number) {  
        super(theX, theY);  
    }  
  
    // get/set accessors ...  
}
```



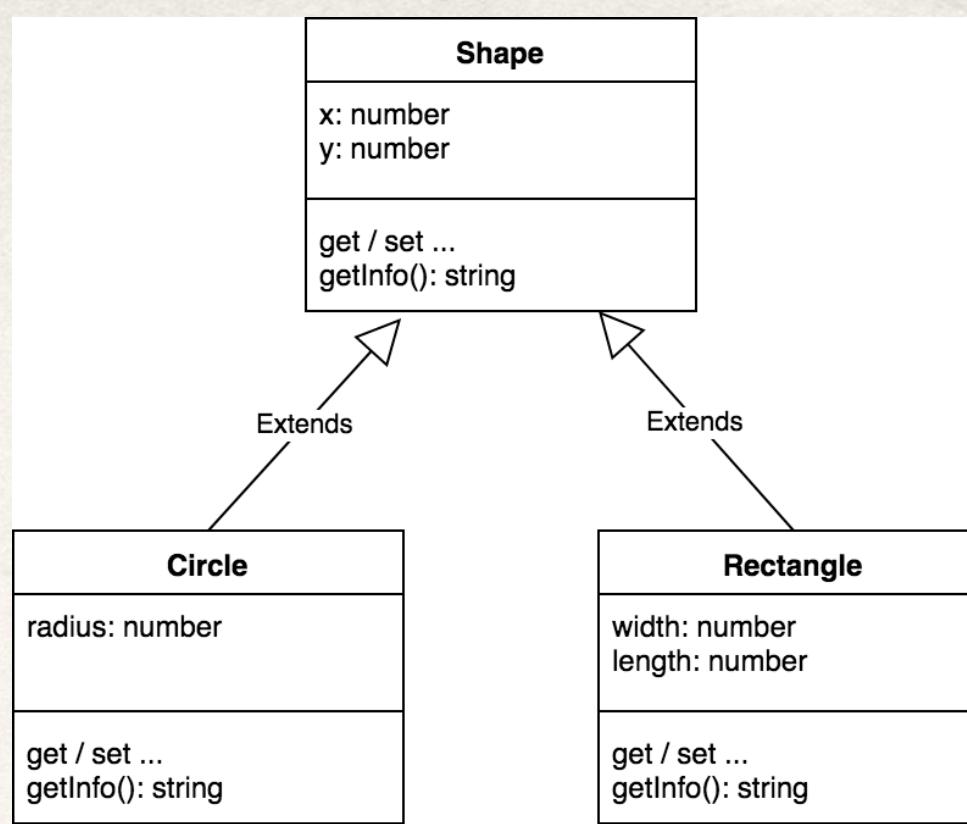
# Inheritance Example

File: Shape.ts

```
export class Shape {  
  
    constructor(private _x: number, private _y: number) {  
    }  
  
    // get/set accessors ...  
  
    getInfo(): string {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

File: Circle.ts

```
import { Shape } from './Shape';  
  
export class Circle extends Shape {  
  
    constructor(theX: number, theY: number,  
              private _radius: number) {  
        super(theX, theY);  
    }  
  
    // get/set accessors ...  
  
    getInfo(): string {  
        return super.getInfo() + `, radius=${this._radius}`;  
    }  
}
```



# Inheritance Example

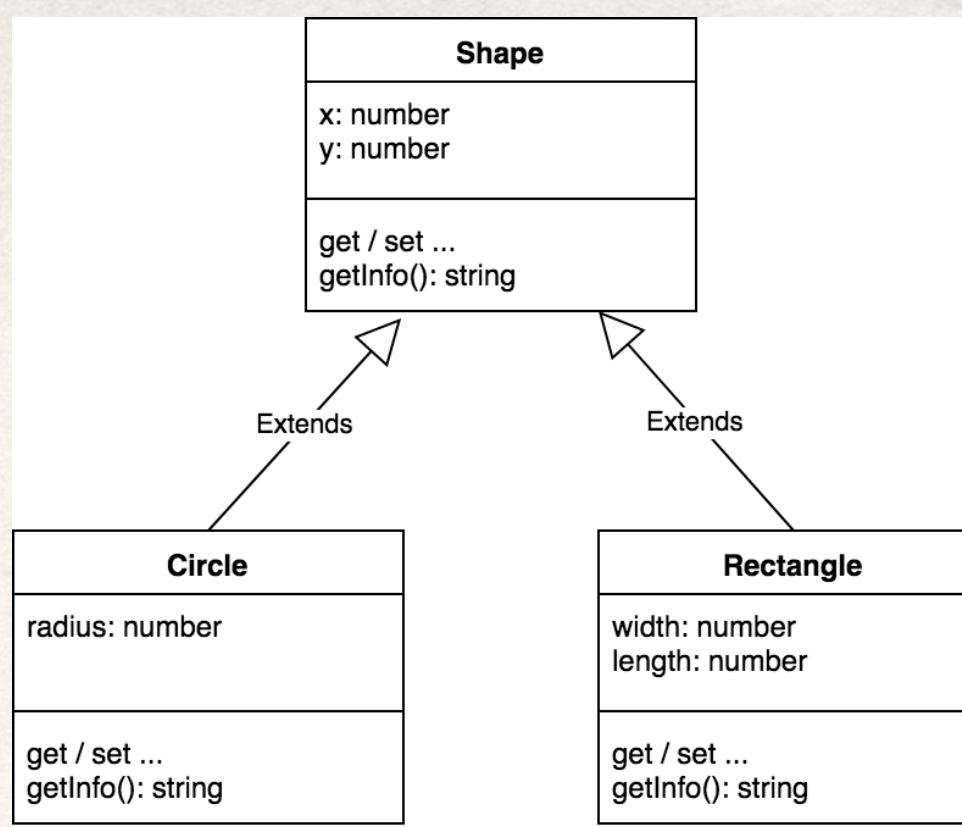
File: Shape.ts

```
export class Shape {  
  
    constructor(private _x: number, private _y: number) {  
    }  
  
    // get/set accessors ...  
  
    getInfo(): string {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

Override the  
getInfo() method

File: Circle.ts

```
import { Shape } from './Shape';  
  
export class Circle extends Shape {  
  
    constructor(theX: number, theY: number,  
              private _radius: number) {  
        super(theX, theY);  
    }  
  
    // get/set accessors ...  
  
    getInfo(): string {  
        return super.getInfo() + `, radius=${this._radius}`;  
    }  
}
```



# Inheritance Example

File: Shape.ts

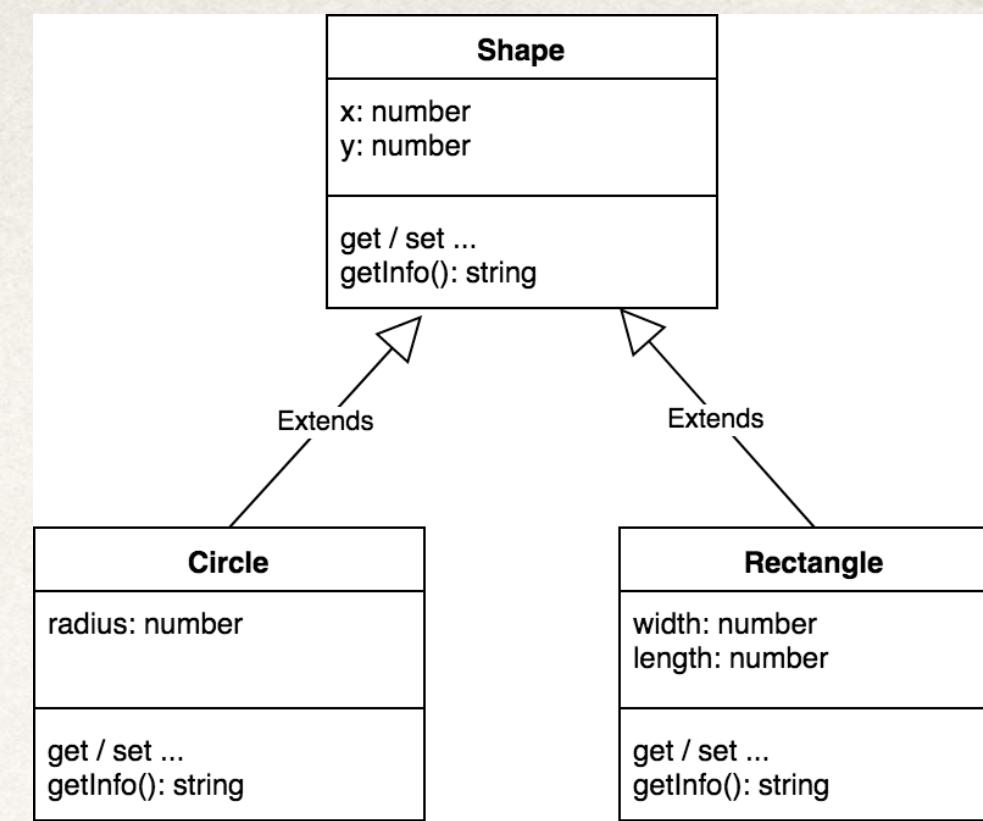
```
export class Shape {  
  
    constructor(private _x: number, private _y: number) {  
    }  
  
    // get/set accessors ...  
  
    getInfo(): string {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

Override the  
getInfo() method

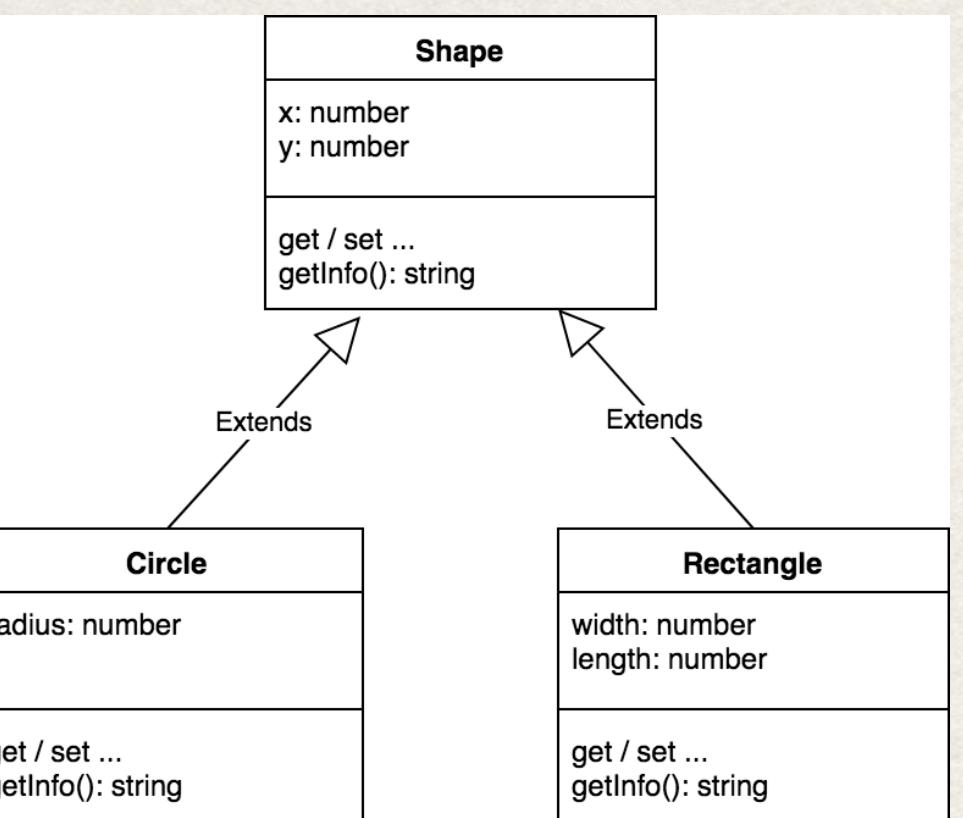
File: Circle.ts

```
import { Shape } from './Shape';  
  
export class Circle extends Shape {  
  
    constructor(theX: number, theY: number,  
              private _radius: number) {  
        super(theX, theY);  
    }  
  
    // get/set accessors ...  
  
    getInfo(): string {  
        return super.getInfo() + ', radius=${this._radius}';  
    }  
}
```

Call superclass  
method



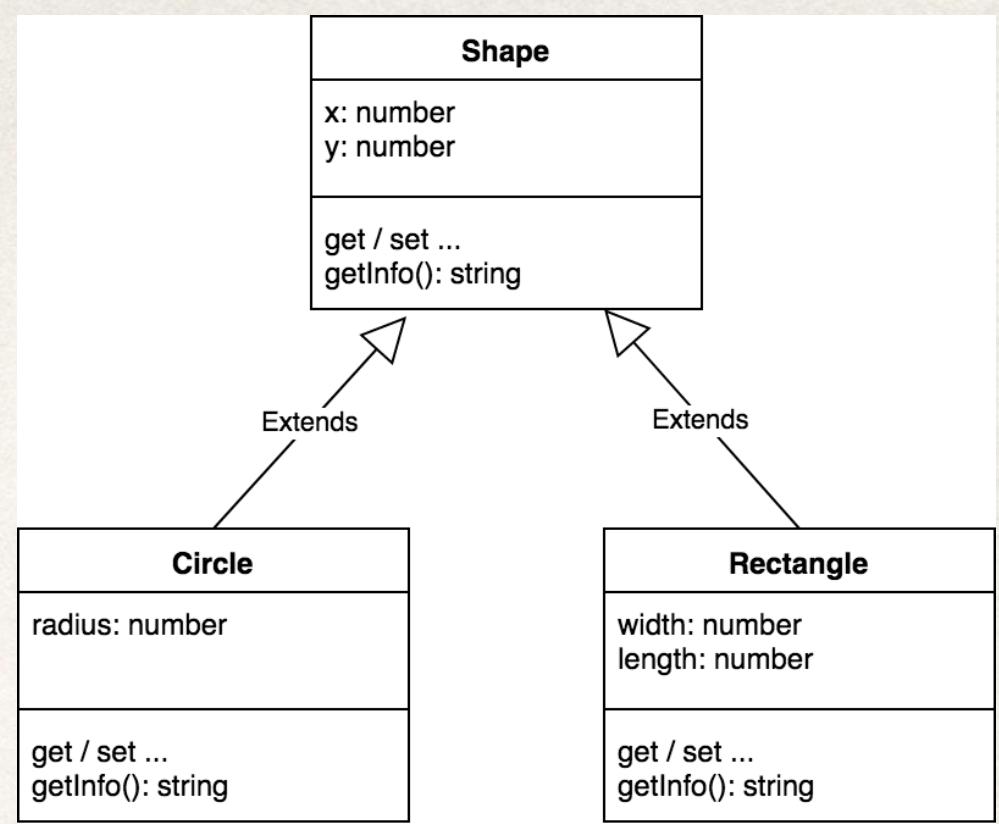
# Creating a main app



# Creating a main app

File: Shape.ts

```
export class Shape {  
    ...  
    getInfo() {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```



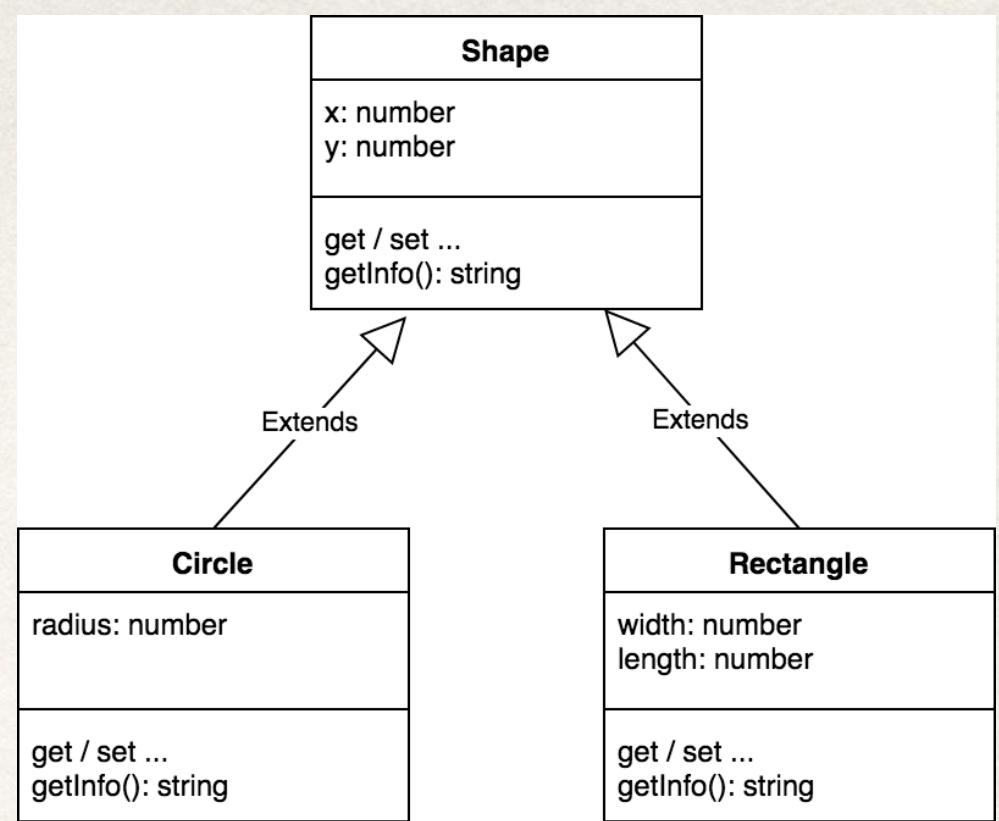
# Creating a main app

File: Shape.ts

```
export class Shape {  
    ...  
    getInfo() {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

File: Circle.ts

```
import { Shape } from './Shape';  
  
export class Circle extends Shape {  
    ...  
    getInfo() {  
        return super.getInfo() + `, radius=${this._radius}`;  
    }  
}
```



# Creating a main app

File: Shape.ts

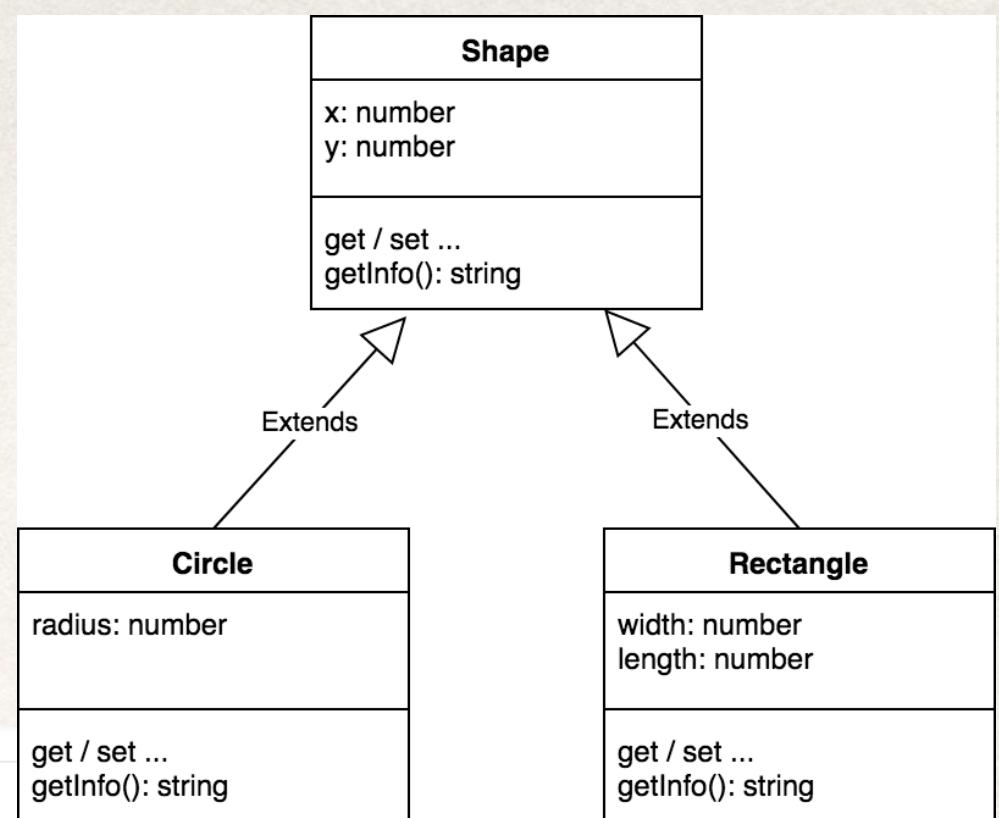
```
export class Shape {  
    ...  
    getInfo() {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

File: Circle.ts

```
import { Shape } from './Shape';  
  
export class Circle extends Shape {  
    ...  
    getInfo() {  
        return super.getInfo() + `, radius=${this._radius}`;  
    }  
}
```

File: Driver.ts

```
import { Shape } from './Shape';  
import { Circle } from './Circle';
```



# Creating a main app

File: Shape.ts

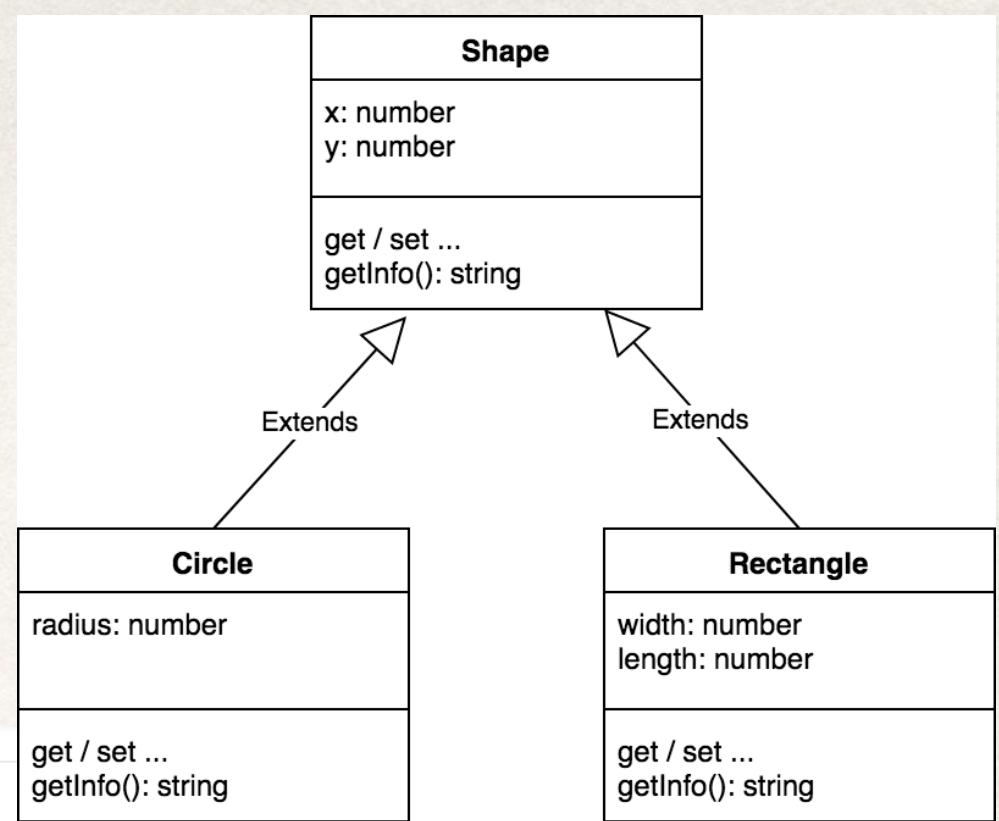
```
export class Shape {  
    ...  
    getInfo() {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

File: Circle.ts

```
import { Shape } from './Shape';  
  
export class Circle extends Shape {  
    ...  
    getInfo() {  
        return super.getInfo() + `, radius=${this._radius}`;  
    }  
}
```

File: Driver.ts

```
import { Shape } from './Shape';  
import { Circle } from './Circle';  
  
let myShape = new Shape(10, 15);
```



# Creating a main app

File: Shape.ts

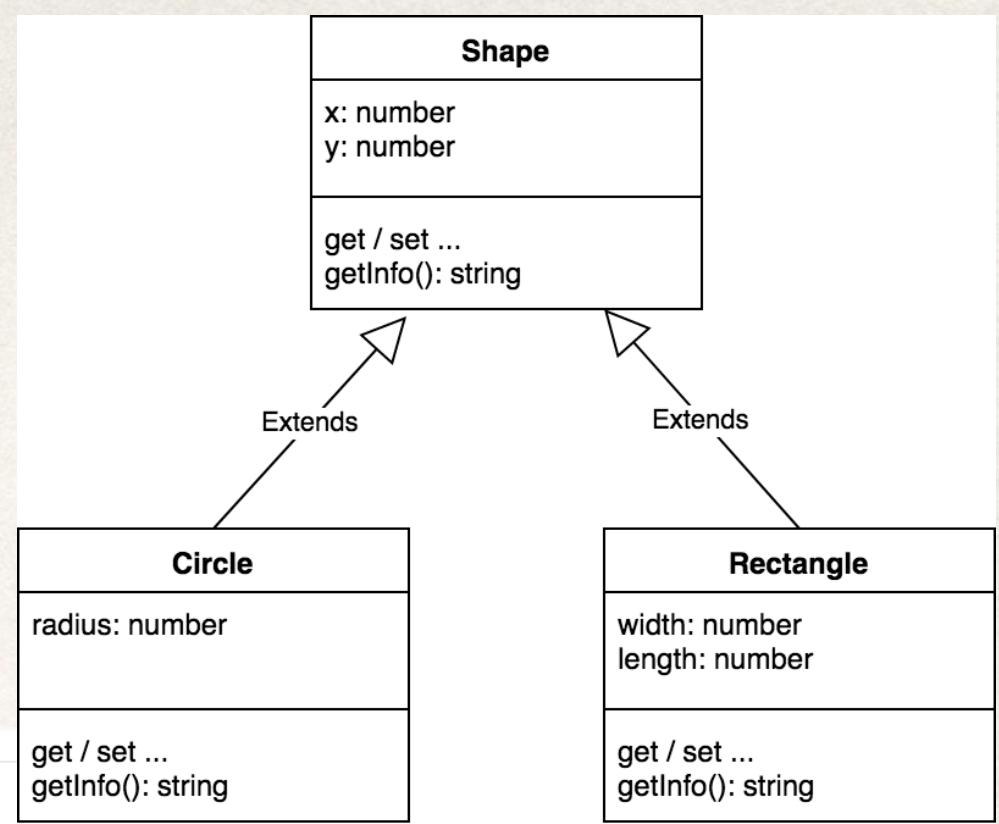
```
export class Shape {  
    ...  
    getInfo() {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

File: Circle.ts

```
import { Shape } from './Shape';  
  
export class Circle extends Shape {  
    ...  
    getInfo() {  
        return super.getInfo() + `, radius=${this._radius}`;  
    }  
}
```

File: Driver.ts

```
import { Shape } from './Shape';  
import { Circle } from './Circle';  
  
let myShape = new Shape(10, 15);  
console.log(myShape.getInfo());
```



# Creating a main app

File: Shape.ts

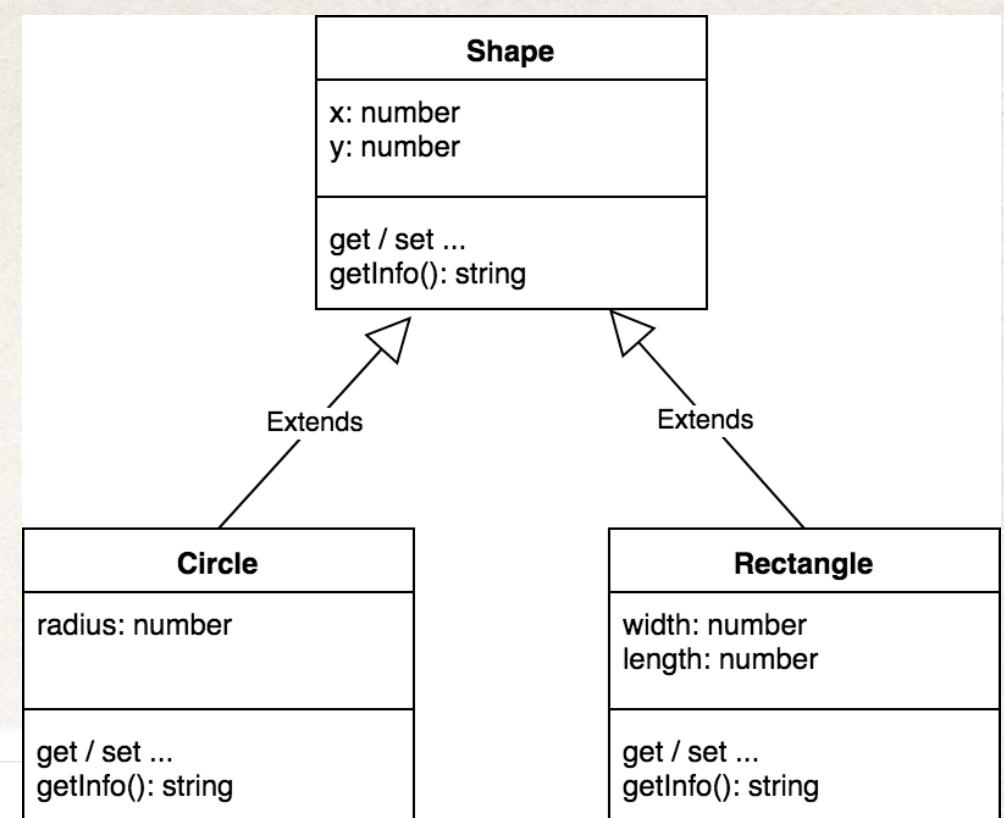
```
export class Shape {  
    ...  
    getInfo() {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

File: Circle.ts

```
import { Shape } from './Shape';  
  
export class Circle extends Shape {  
    ...  
    getInfo() {  
        return super.getInfo() + `, radius=${this._radius}`;  
    }  
}
```

File: Driver.ts

```
import { Shape } from './Shape';  
import { Circle } from './Circle';  
  
let myShape = new Shape(10, 15);  
console.log(myShape.getInfo());  
  
let myCircle = new Circle(5, 10, 20);
```



# Creating a main app

File: Shape.ts

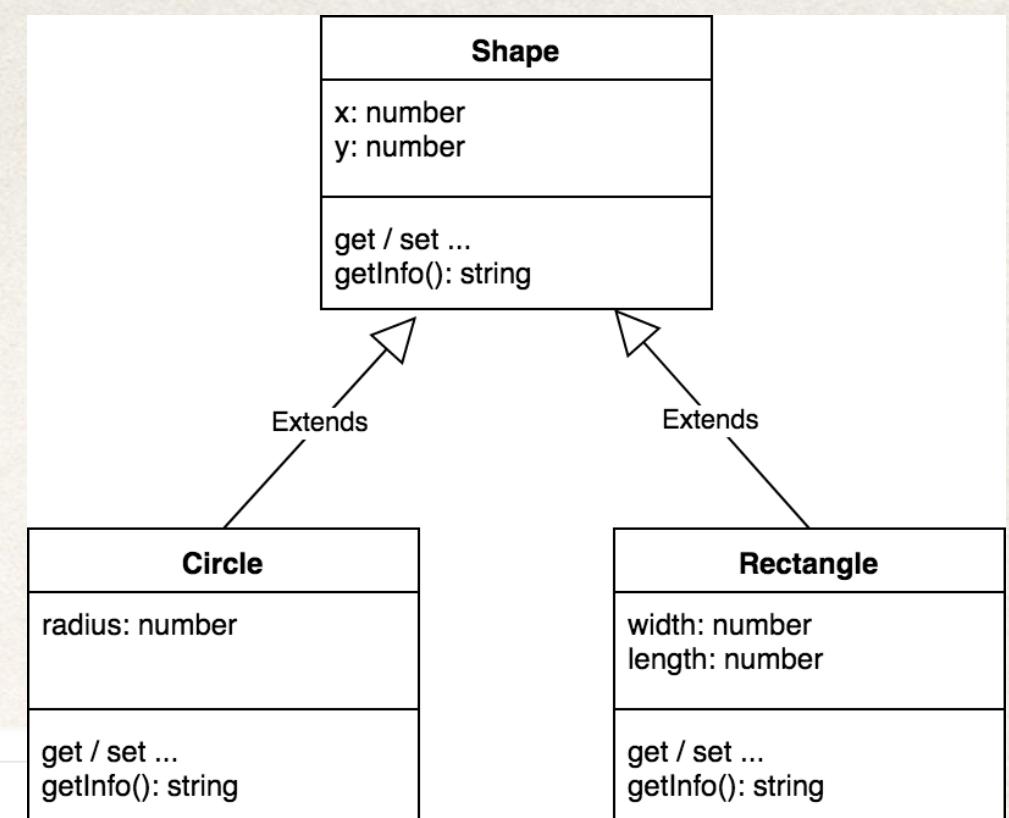
```
export class Shape {  
    ...  
    getInfo() {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

File: Circle.ts

```
import { Shape } from './Shape';  
  
export class Circle extends Shape {  
    ...  
    getInfo() {  
        return super.getInfo() + ', radius=${this._radius}';  
    }  
}
```

File: Driver.ts

```
import { Shape } from './Shape';  
import { Circle } from './Circle';  
  
let myShape = new Shape(10, 15);  
console.log(myShape.getInfo());  
  
let myCircle = new Circle(5, 10, 20);  
console.log(myCircle.getInfo());
```



# Creating a main app

File: Shape.ts

```
export class Shape {  
    ...  
    getInfo() {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

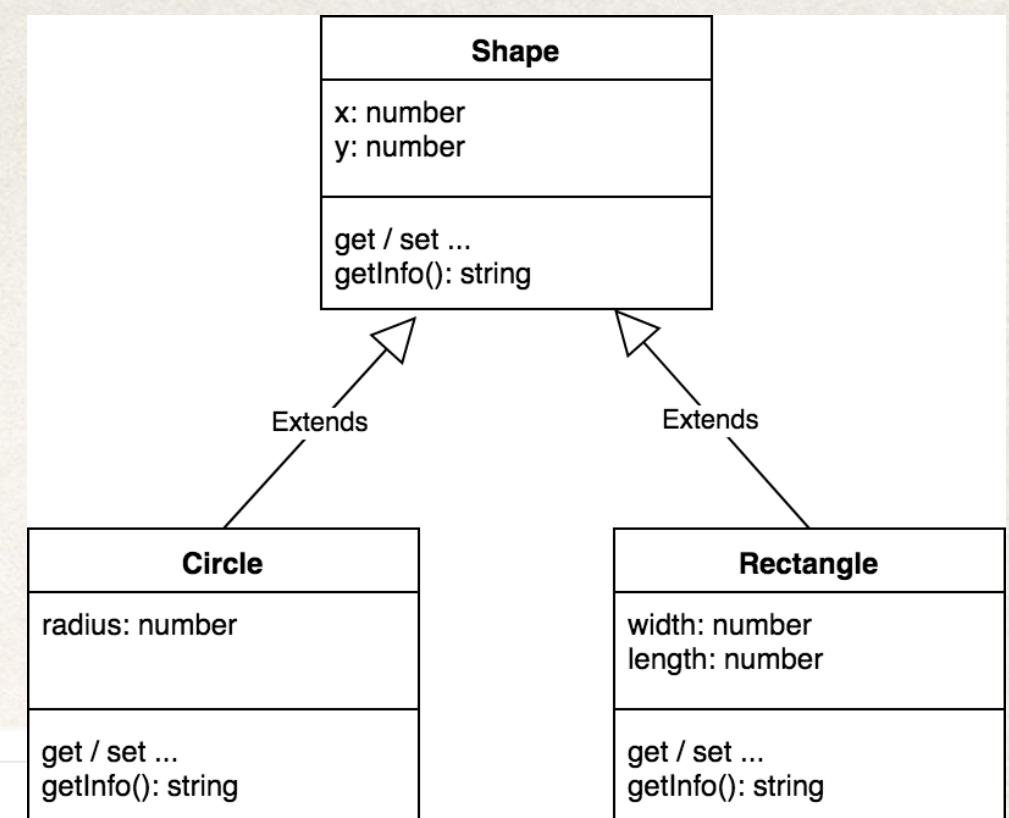
File: Circle.ts

```
import { Shape } from './Shape';  
  
export class Circle extends Shape {  
    ...  
    getInfo() {  
        return super.getInfo() + ', radius=${this._radius}';  
    }  
}
```

File: Driver.ts

```
import { Shape } from './Shape';  
import { Circle } from './Circle';  
  
let myShape = new Shape(10, 15);  
console.log(myShape.getInfo());  
  
let myCircle = new Circle(5, 10, 20);  
console.log(myCircle.getInfo());
```

x=10, y=15  
x=5, y=10, radius=20



# Creating a main app

File: Shape.ts

```
export class Shape {  
    ...  
    getInfo() {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

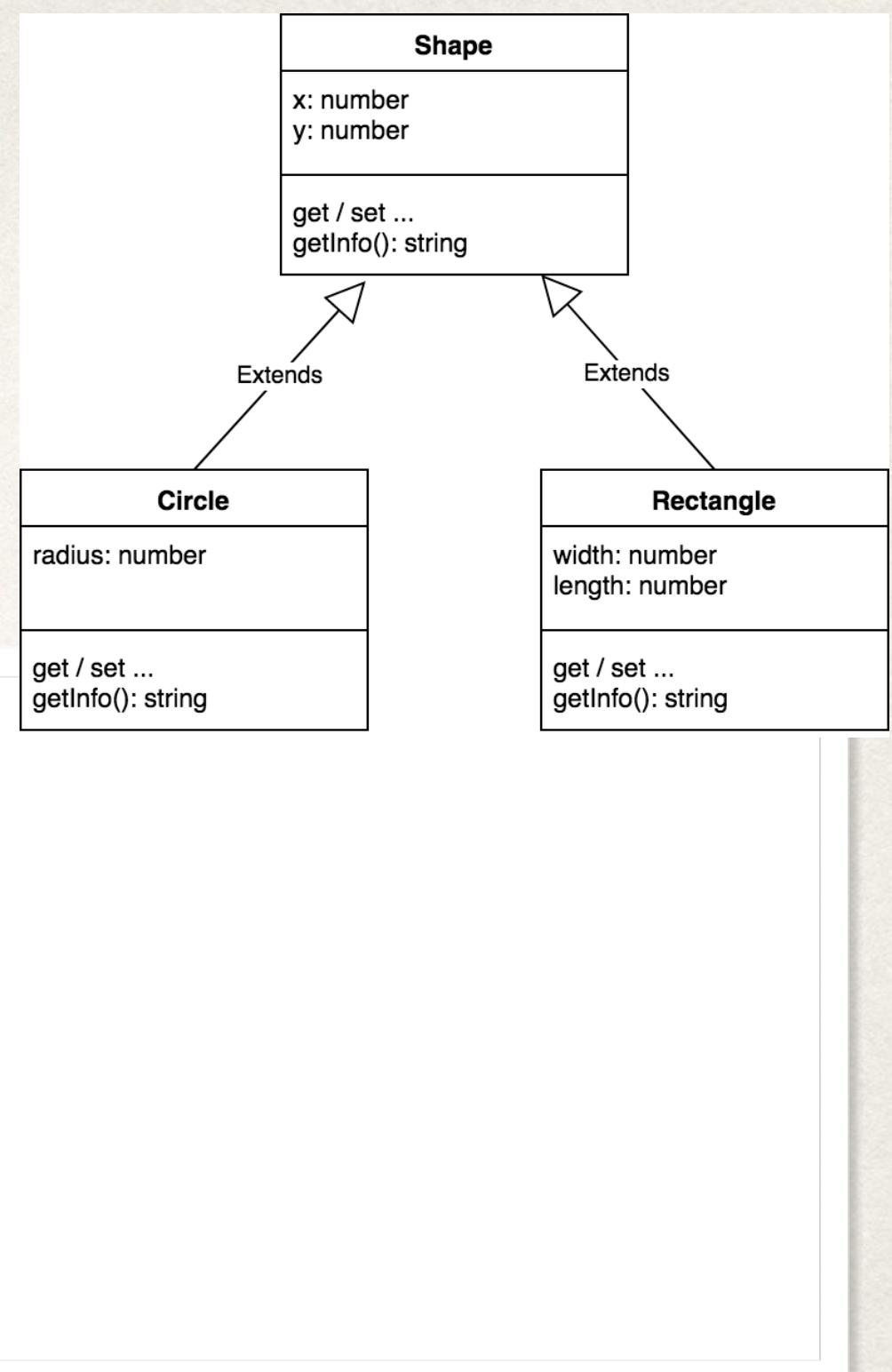
File: Circle.ts

```
import { Shape } from './Shape';  
  
export class Circle extends Shape {  
    ...  
    getInfo() {  
        return super.getInfo() + ', radius=${this._radius}';  
    }  
}
```

File: Driver.ts

```
import { Shape } from './Shape';  
import { Circle } from './Circle';  
  
let myShape = new Shape(10, 15);  
console.log(myShape.getInfo());  
  
let myCircle = new Circle(5, 10, 20);  
console.log(myCircle.getInfo());
```

x=10, y=15  
x=5, y=10, radius=20



# Creating a main app

File: Shape.ts

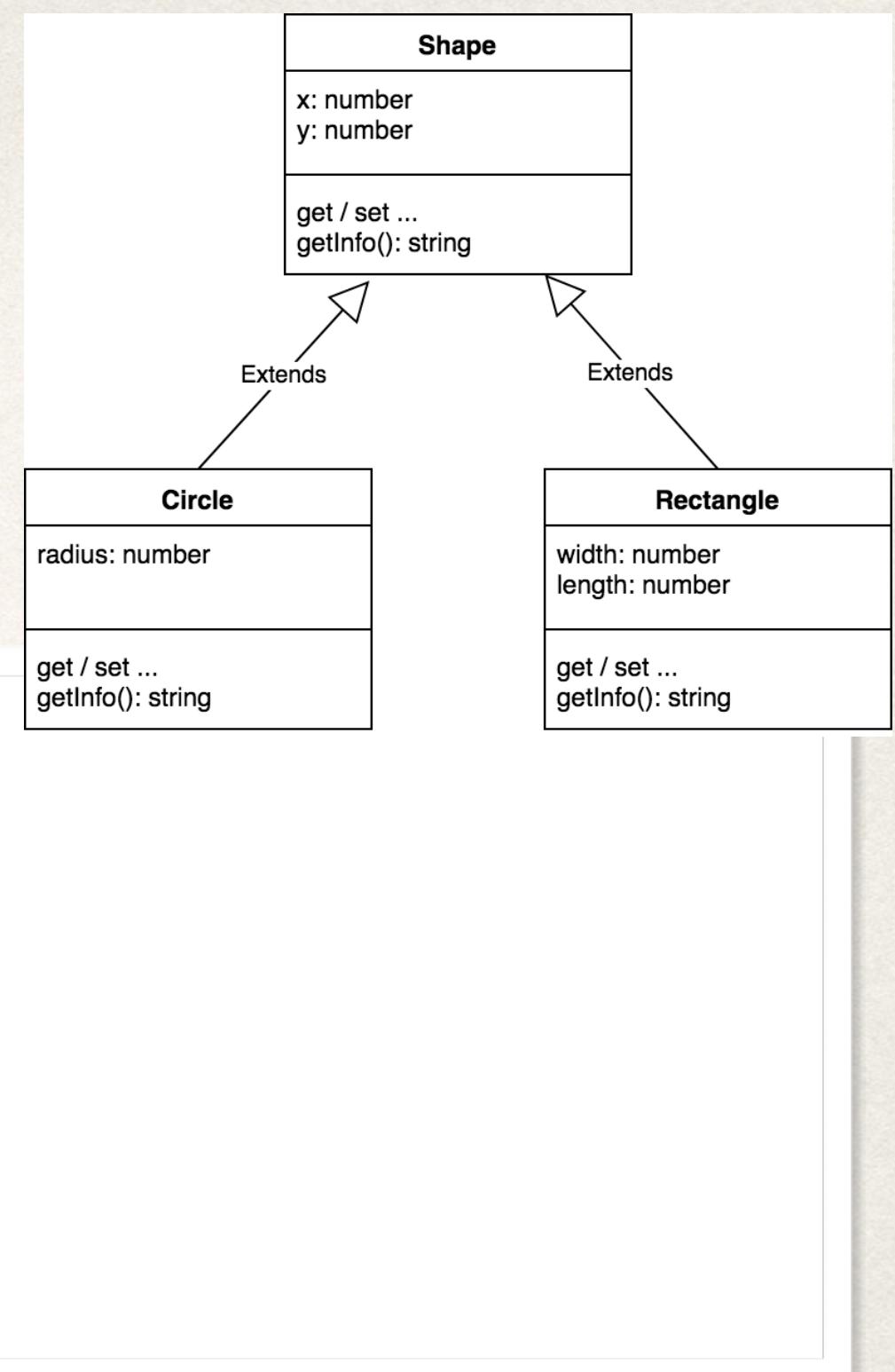
```
export class Shape {  
    ...  
    getInfo() {  
        return `x=${this._x}, y=${this._y}`;  
    }  
}
```

File: Circle.ts

```
import { Shape } from './Shape';  
  
export class Circle extends Shape {  
    ...  
    getInfo() {  
        return super.getInfo() + ', radius=${this._radius}';  
    }  
}
```

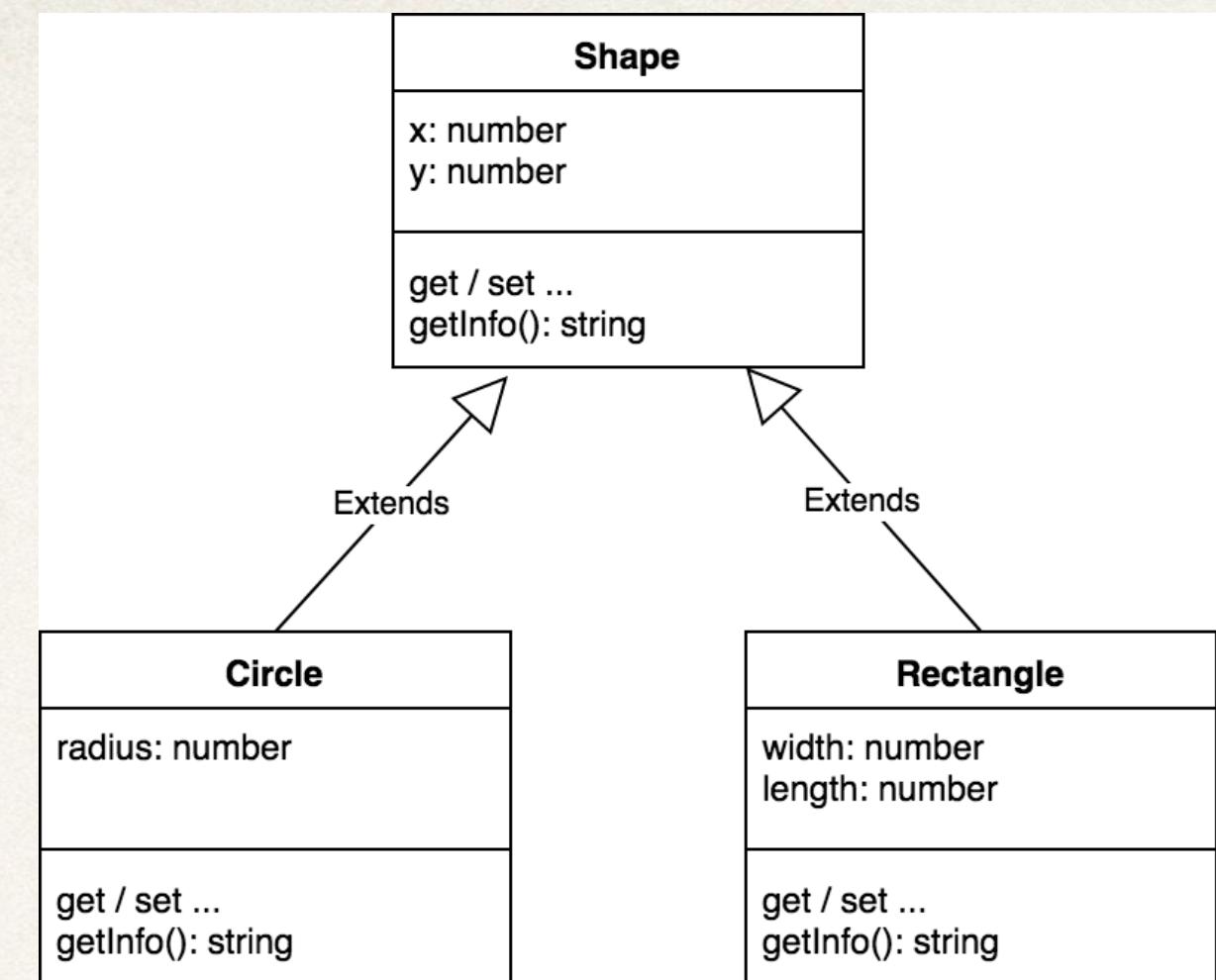
File: Driver.ts

```
import { Shape } from './Shape';  
import { Circle } from './Circle';  
  
let myShape = new Shape(10, 15);  
console.log(myShape.getInfo());  
  
let myCircle = new Circle(5, 10, 20);  
console.log(myCircle.getInfo());
```



# Rectangle

File: Rectangle.ts

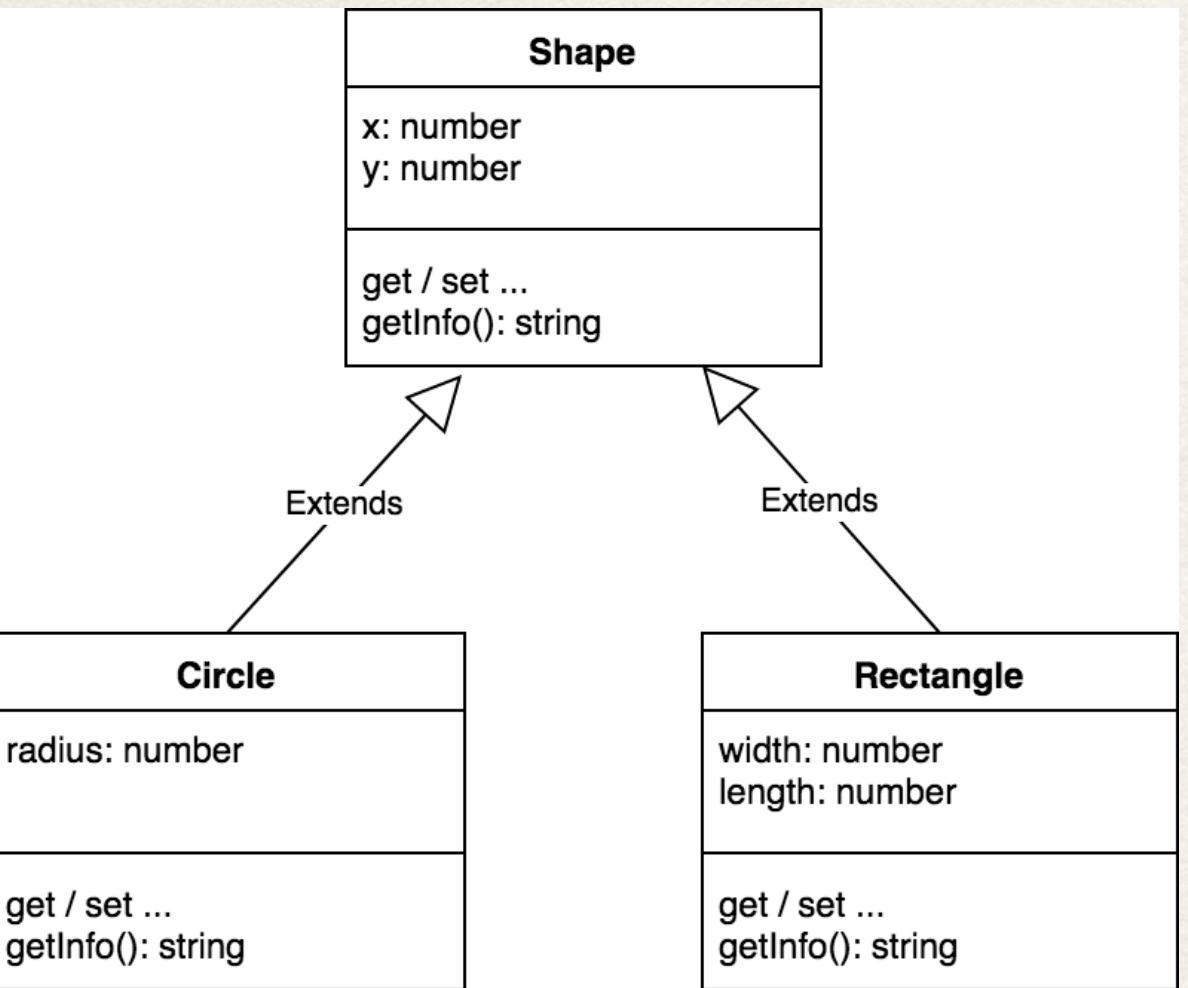


# Rectangle

File: Rectangle.ts

```
import { Shape } from './Shape';

export class Rectangle extends Shape {
```



# Rectangle

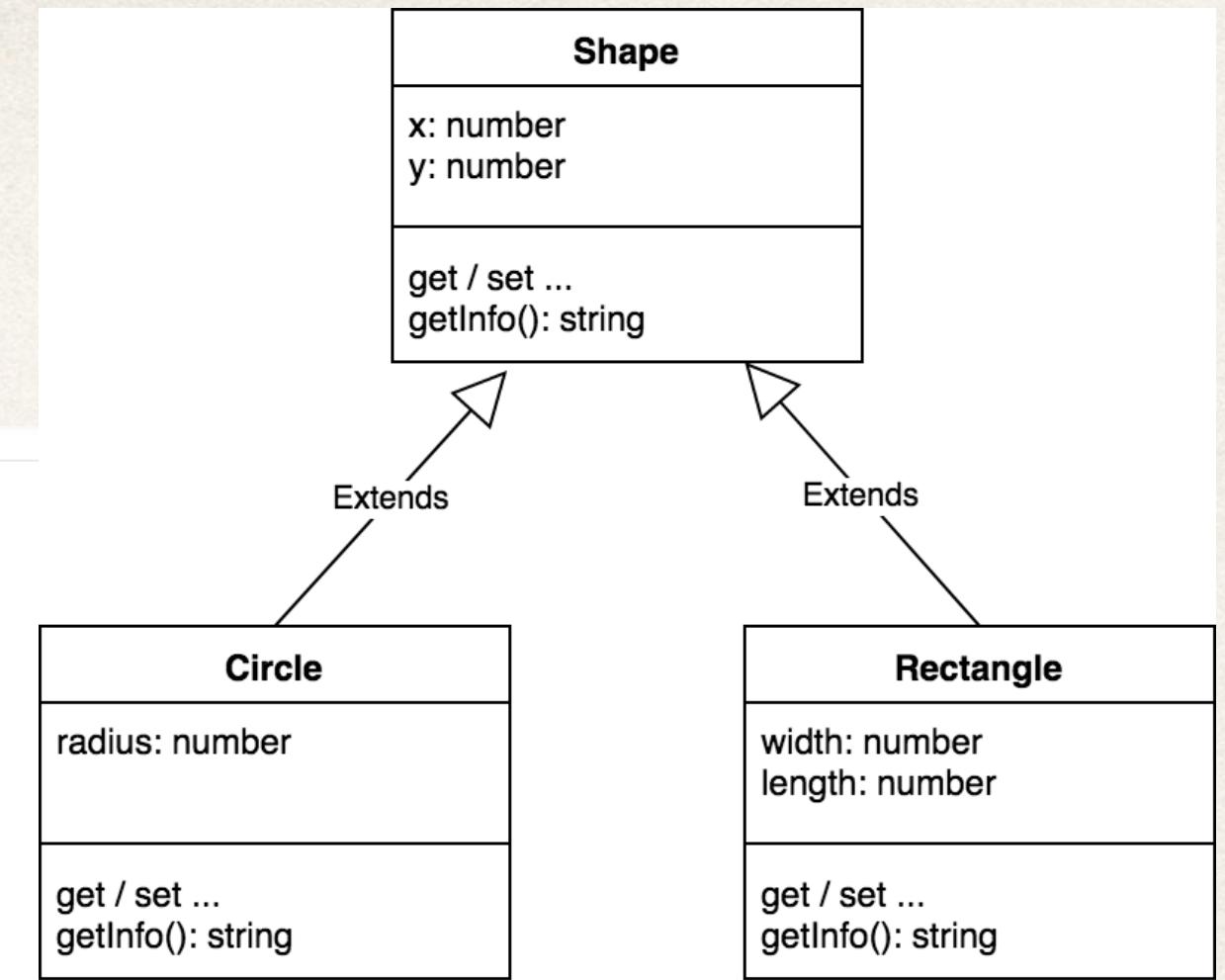
File: Rectangle.ts

```
import { Shape } from './Shape';

export class Rectangle extends Shape {

    constructor(theX: number, theY: number,
        private _width: number, private _length: number) {

        super(theX, theY);
    }
}
```



# Rectangle

File: Rectangle.ts

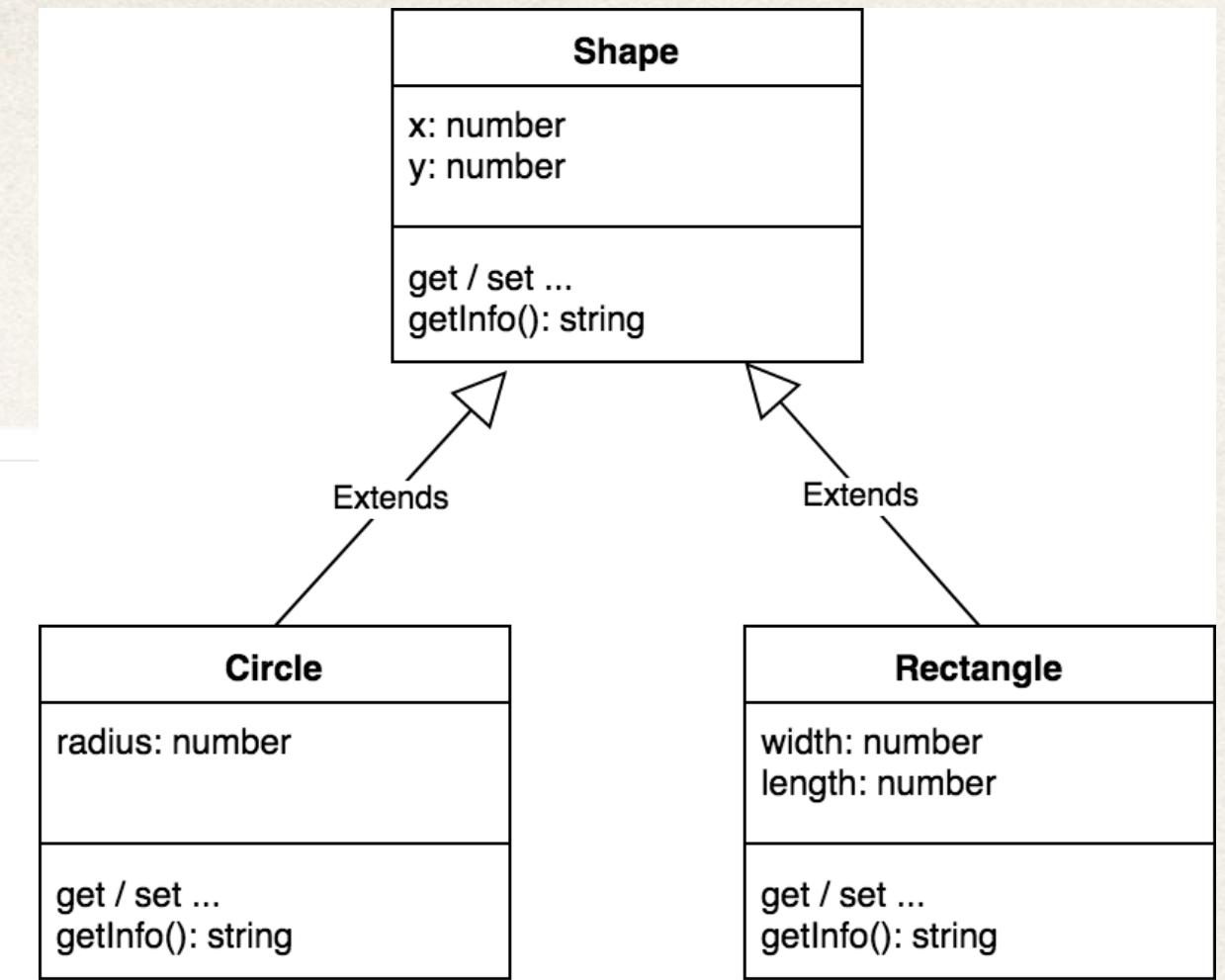
```
import { Shape } from './Shape';

export class Rectangle extends Shape {

    constructor(theX: number, theY: number,
        private _width: number, private _length: number) {

        super(theX, theY);
    }
}
```

Regular parameters  
theX and theY



# Rectangle

File: Rectangle.ts

```
import { Shape } from './Shape';

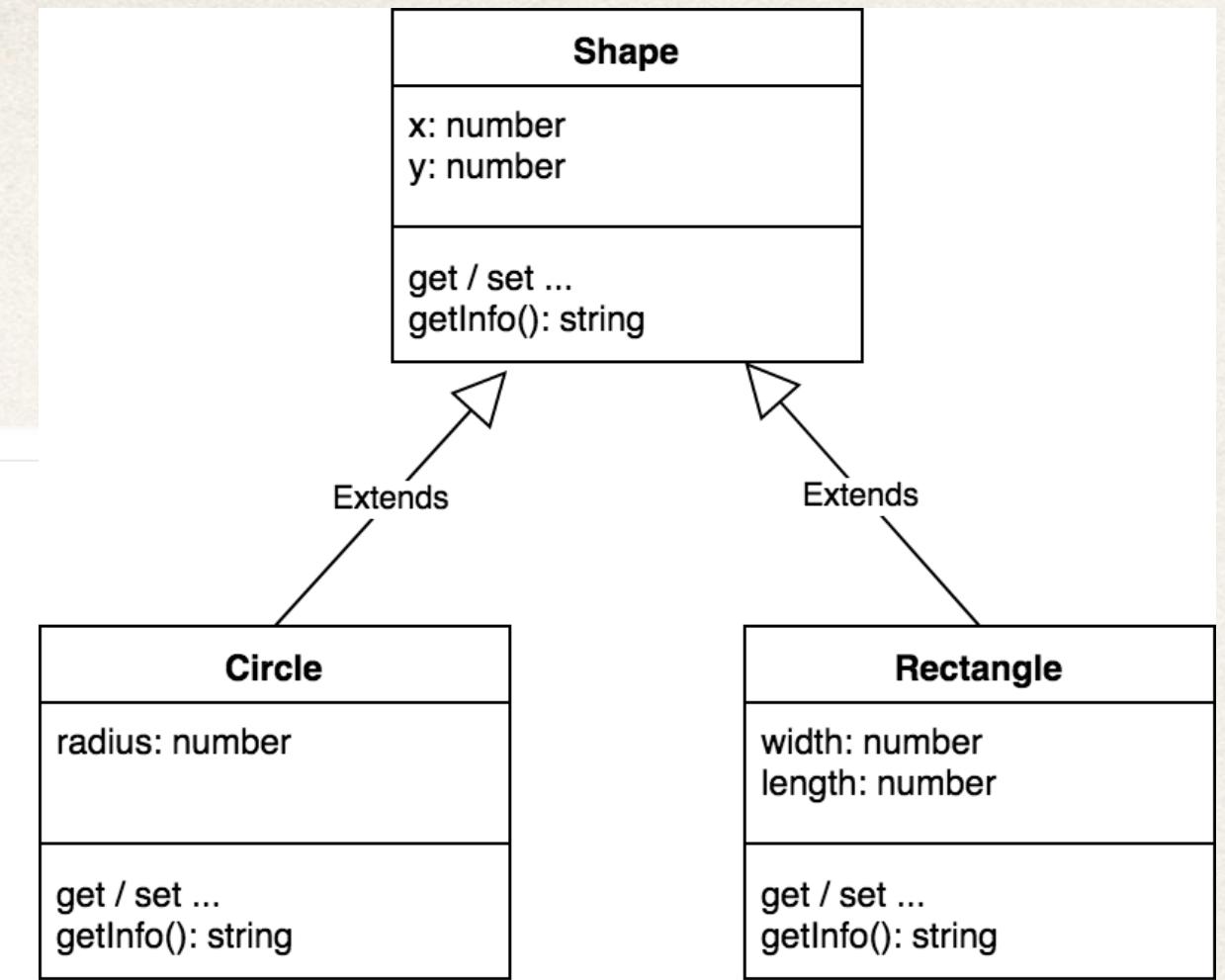
export class Rectangle extends Shape {

    constructor(theX: number, theY: number,
        private _width: number, private _length: number) {

        super(theX, theY);
    }
}
```

Regular parameters  
theX and theY

Parameter Properties  
\_width and \_length



# Rectangle

File: Rectangle.ts

```
import { Shape } from './Shape';

export class Rectangle extends Shape {

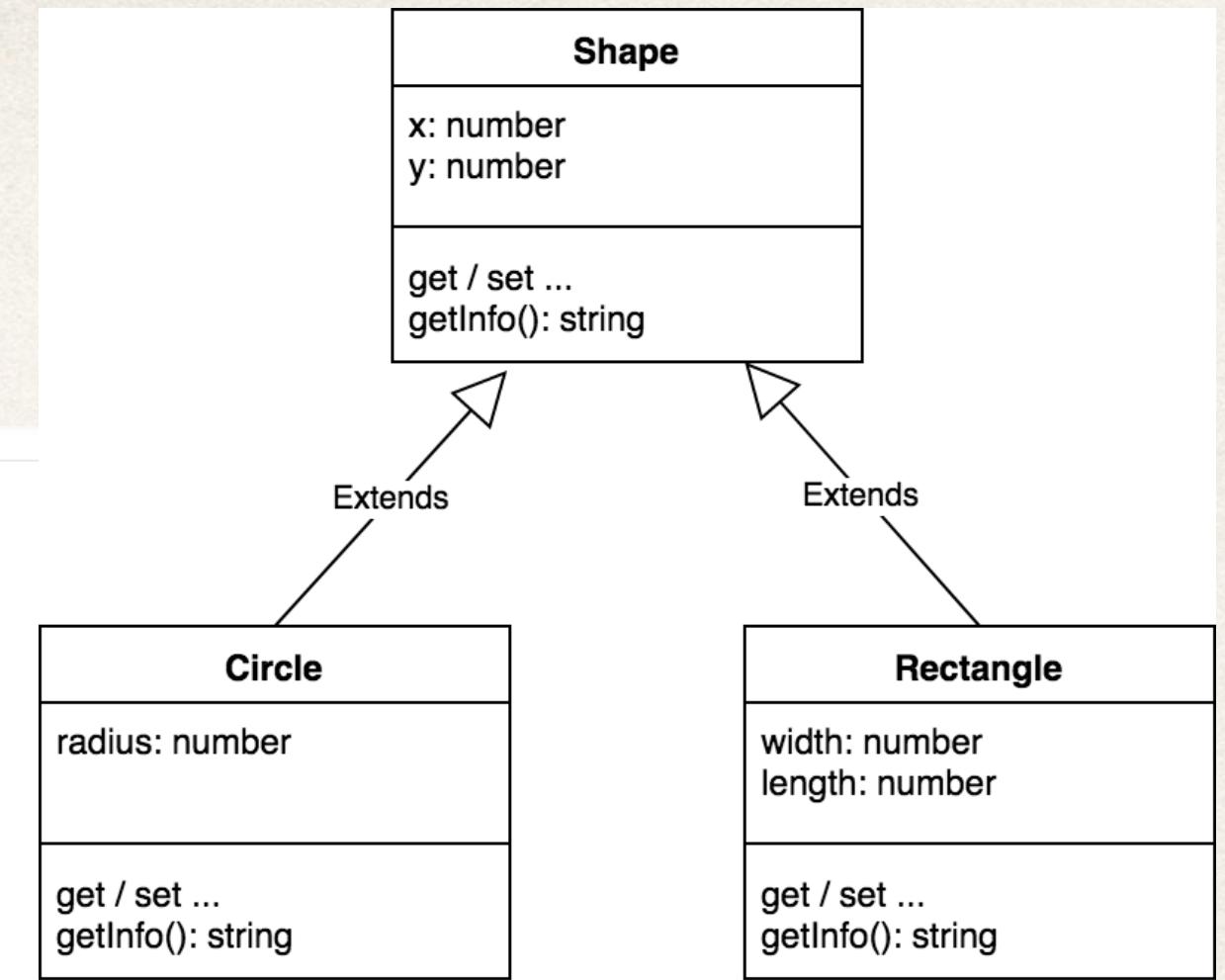
    constructor(theX: number, theY: number,
        private _width: number, private _length: number) {

        super(theX, theY);
    }
}
```

Call superclass  
constructor

Regular parameters  
theX and theY

Parameter Properties  
\_width and \_length



# Rectangle

File: Rectangle.ts

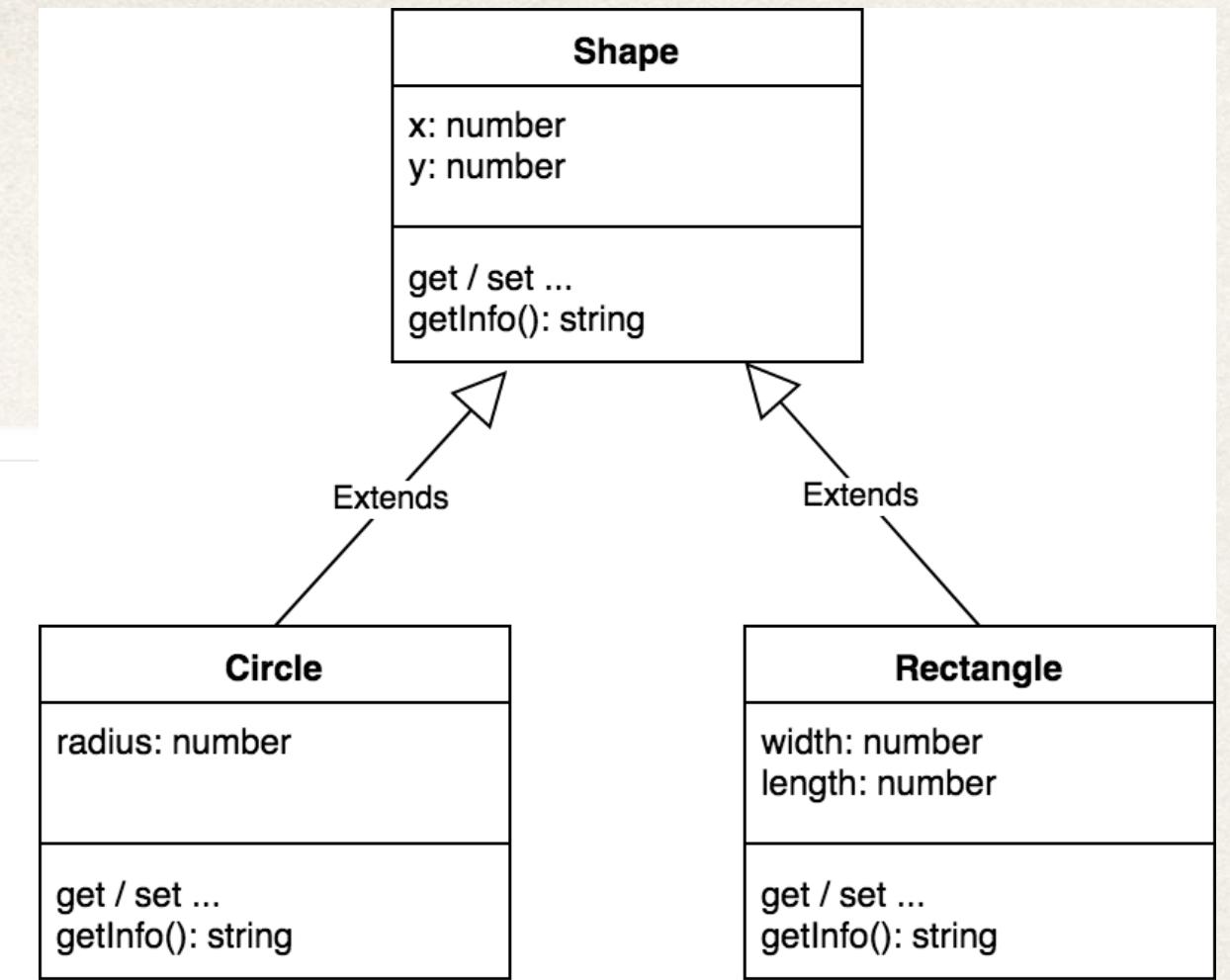
```
import { Shape } from './Shape';

export class Rectangle extends Shape {

    constructor(theX: number, theY: number,
        private _width: number, private _length: number) {

        super(theX, theY);
    }

    // get/set accessors ...
}
```



# Rectangle

File: Rectangle.ts

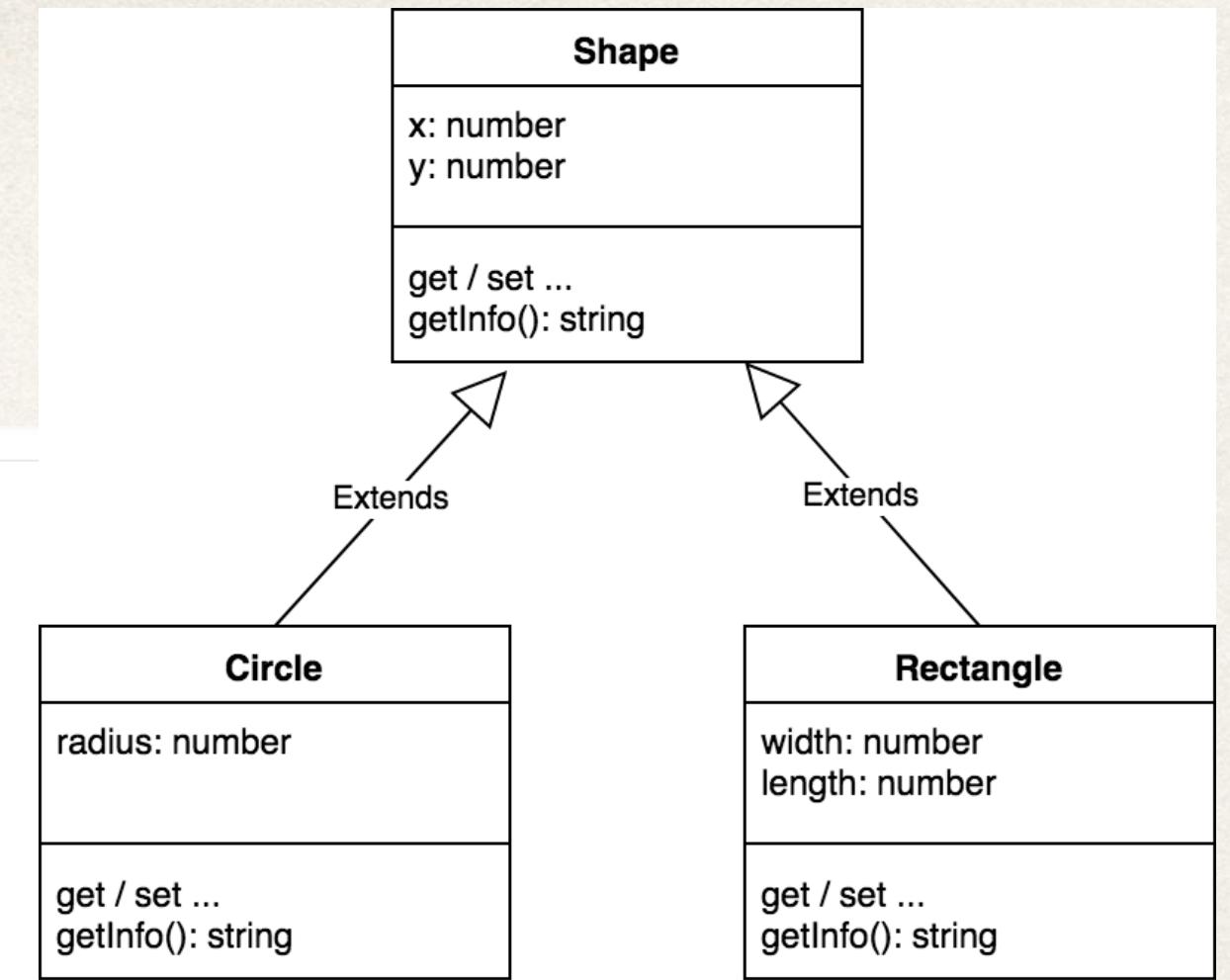
```
import { Shape } from './Shape';

export class Rectangle extends Shape {

    constructor(theX: number, theY: number,
        private _width: number, private _length: number) {

        super(theX, theY);
    }

    // get/set accessors ...
}
```



# Rectangle

File: Rectangle.ts

```
import { Shape } from './Shape';

export class Rectangle extends Shape {

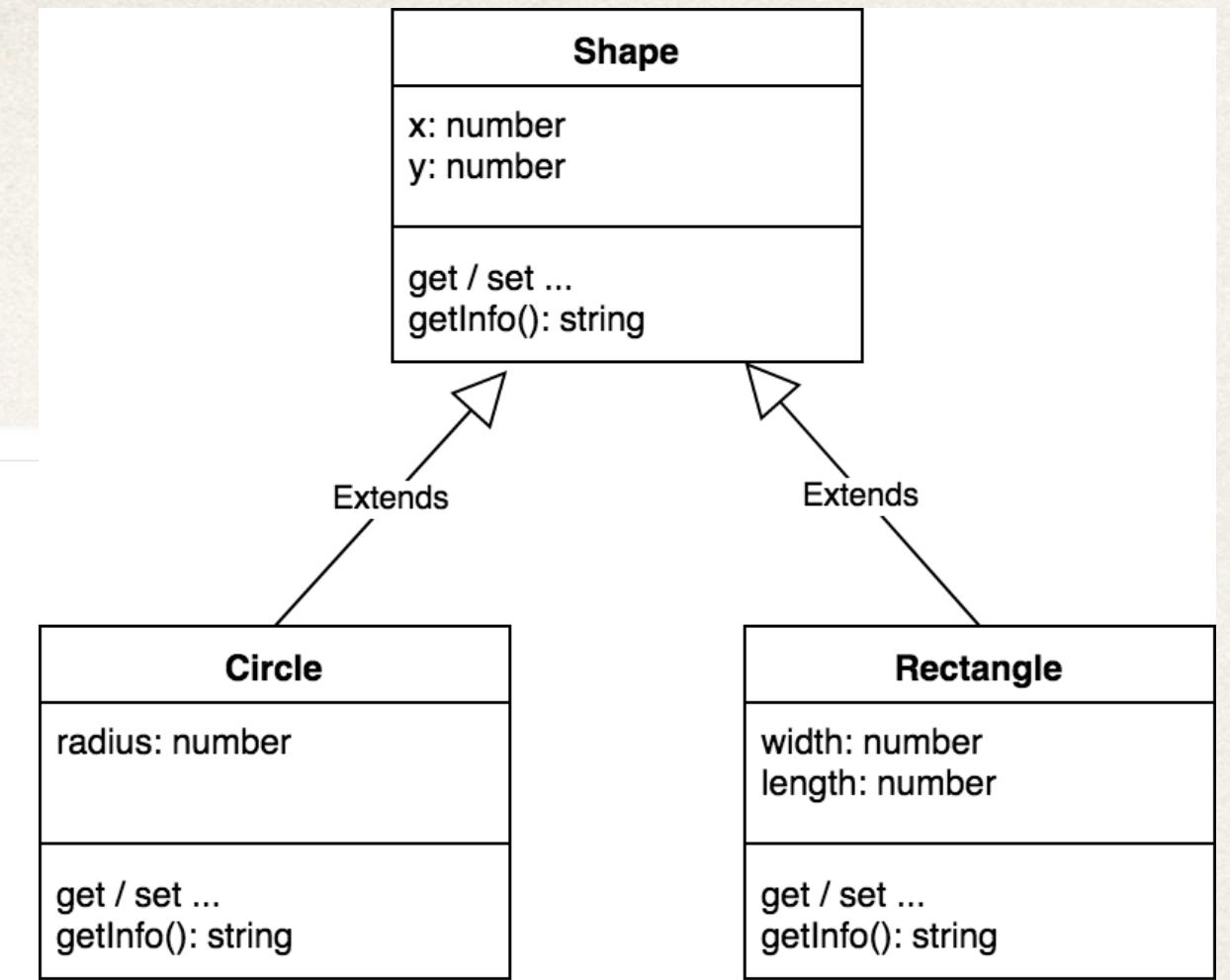
    constructor(theX: number, theY: number,
        private _width: number, private _length: number) {

        super(theX, theY);
    }

    // get/set accessors ...

    getInfo(): string {

```



# Rectangle

File: Rectangle.ts

```
import { Shape } from './Shape';

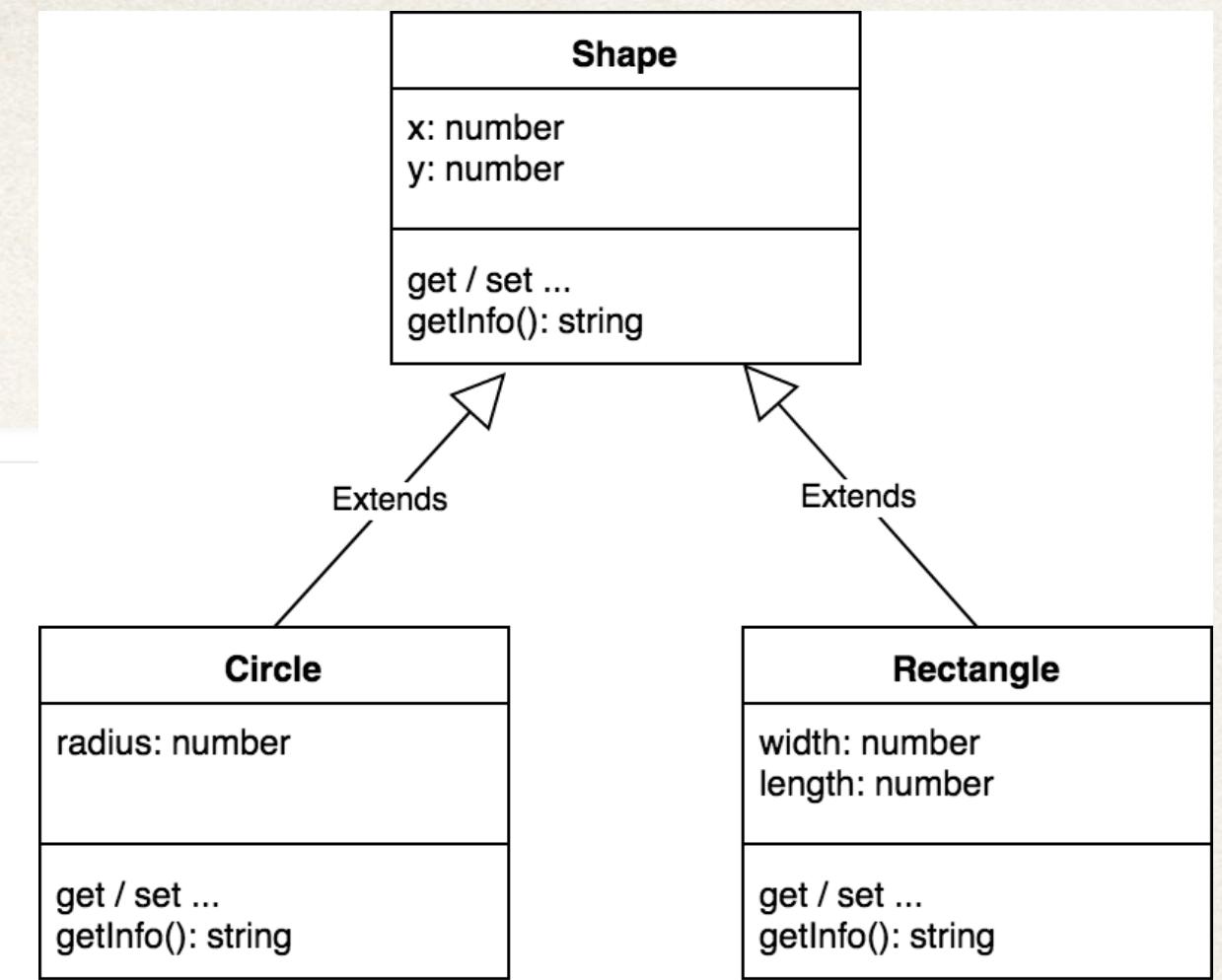
export class Rectangle extends Shape {

    constructor(theX: number, theY: number,
        private _width: number, private _length: number) {

        super(theX, theY);
    }

    // get/set accessors ...

    getInfo(): string {
        return super.getInfo() + `, width=${this._width}, length=${this._length}`;
    }
}
```



# Rectangle

File: Rectangle.ts

```
import { Shape } from './Shape';

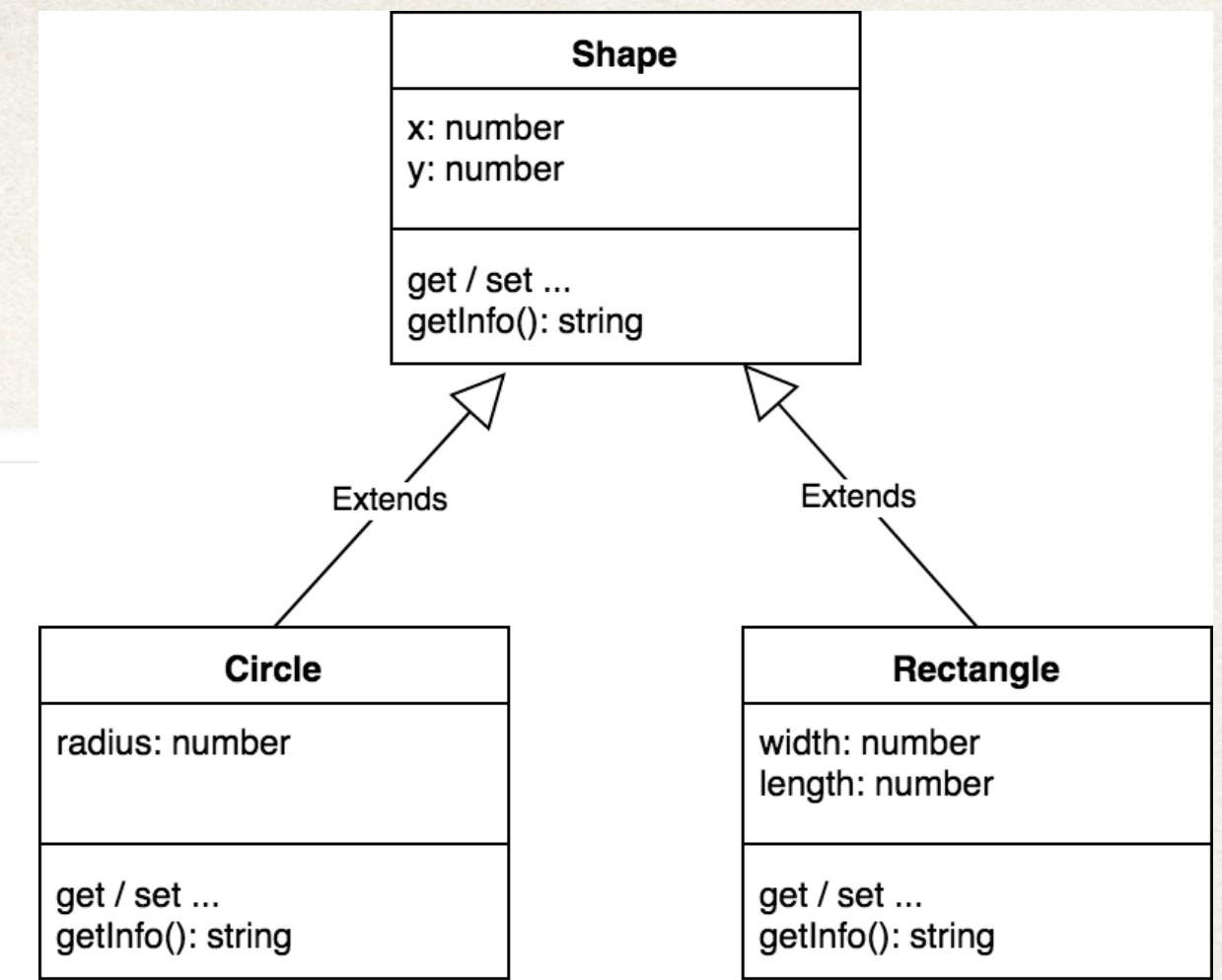
export class Rectangle extends Shape {

    constructor(theX: number, theY: number,
        private _width: number, private _length: number) {

        super(theX, theY);
    }

    // get/set accessors ...

    getInfo(): string {
        return super.getInfo() + `, width=${this._width}, length=${this._length}`;
    }
}
```



# Rectangle

File: Rectangle.ts

```
import { Shape } from './Shape';

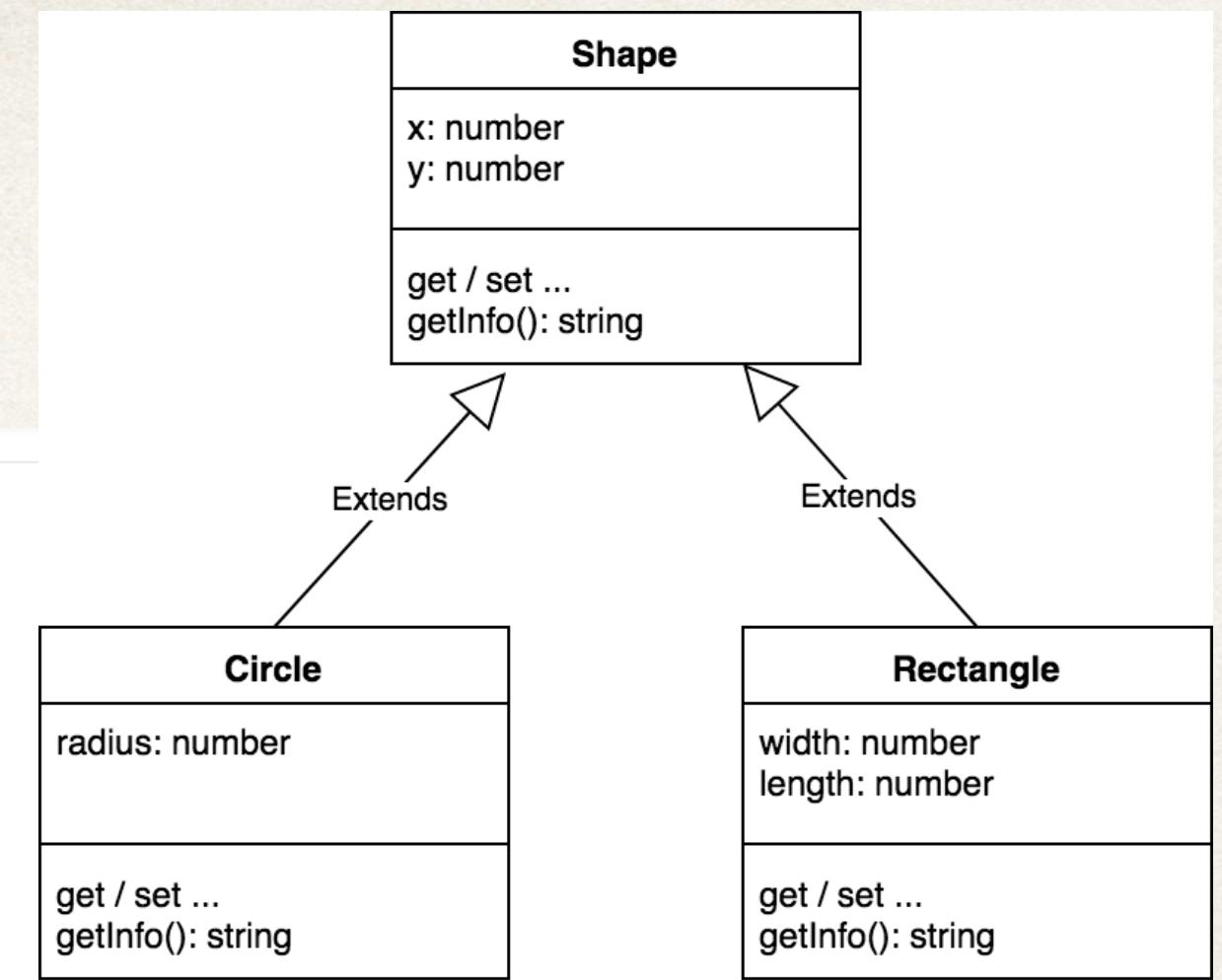
export class Rectangle extends Shape {

    constructor(theX: number, theY: number,
        private _width: number, private _length: number) {

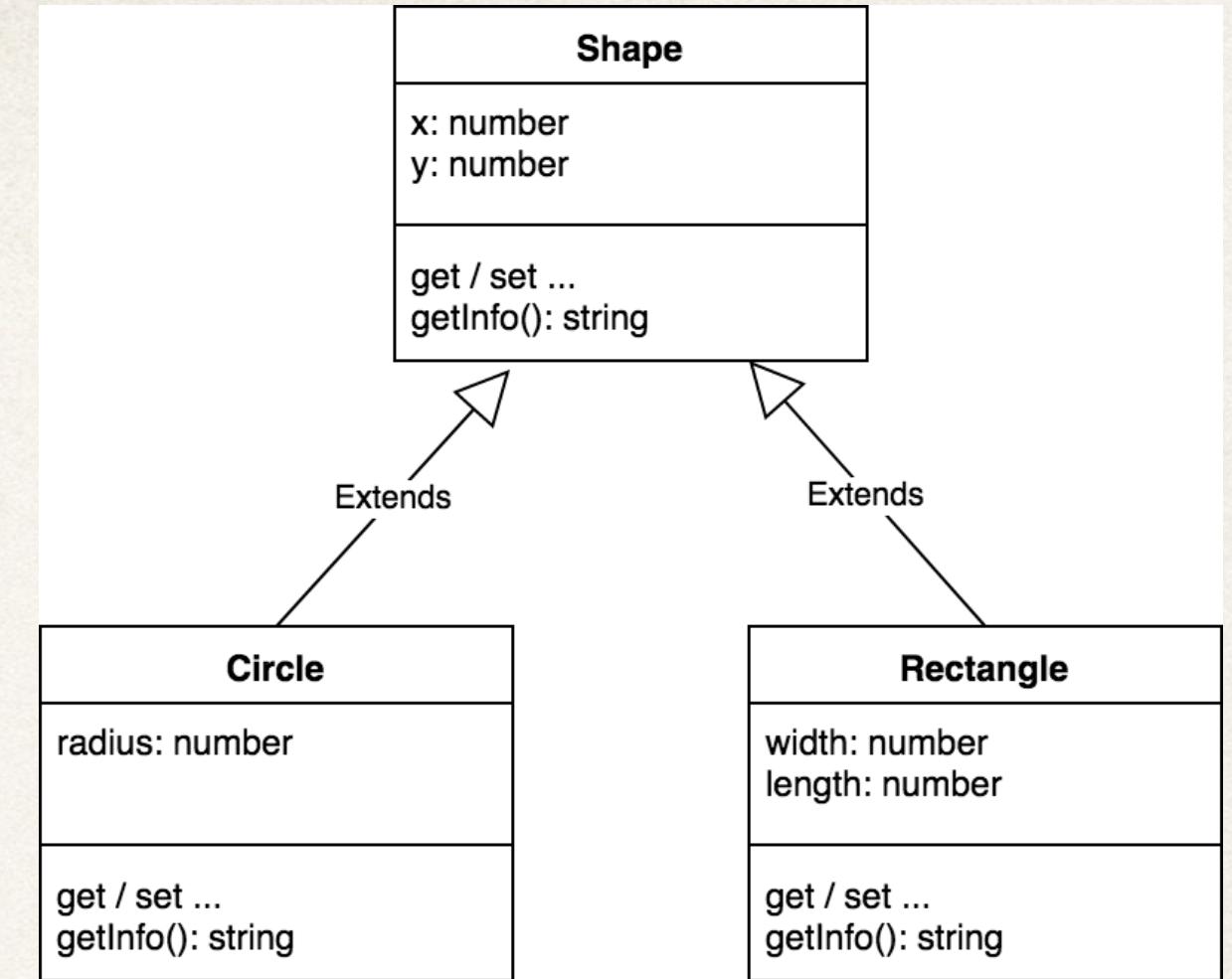
        super(theX, theY);
    }

    // get/set accessors ...

    getInfo(): string {
        return super.getInfo() + `, width=${this._width}, length=${this._length}`;
    }
}
```



# Creating a main app



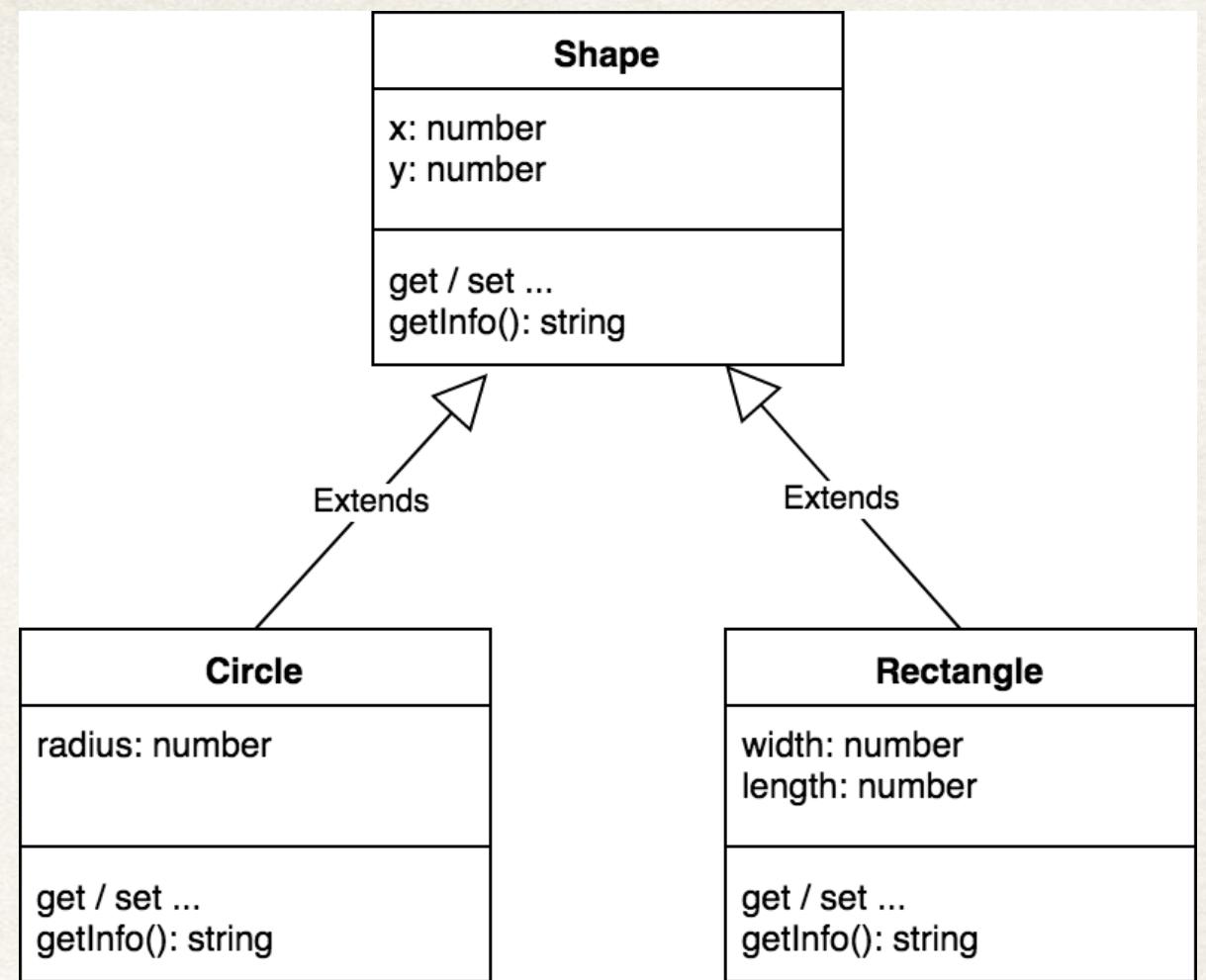
# Creating a main app

File: Driver.ts

```
import { Shape } from './Shape';
import { Circle } from './Circle';
import { Rectangle } from './Rectangle';

let myShape = new Shape(10, 15);
console.log(myShape.getInfo());

let myCircle = new Circle(5, 10, 20);
console.log(myCircle.getInfo());
```



# Creating a main app

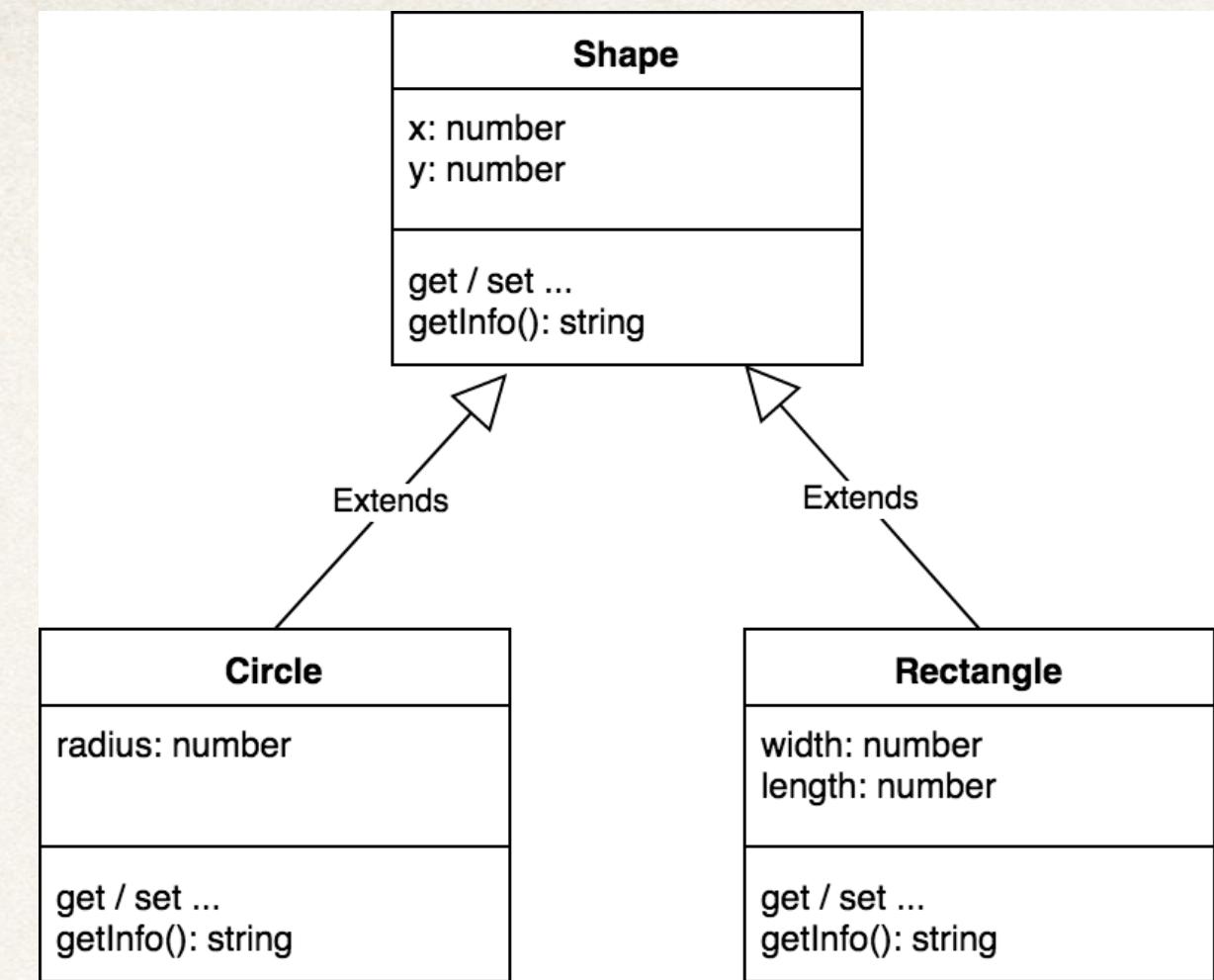
File: Driver.ts

```
import { Shape } from './Shape';
import { Circle } from './Circle';
import { Rectangle } from './Rectangle';

let myShape = new Shape(10, 15);
console.log(myShape.getInfo());

let myCircle = new Circle(5, 10, 20);
console.log(myCircle.getInfo());

let myRectangle = new Rectangle(0, 0, 3, 7);
console.log(myRectangle.getInfo());
```



# Creating a main app

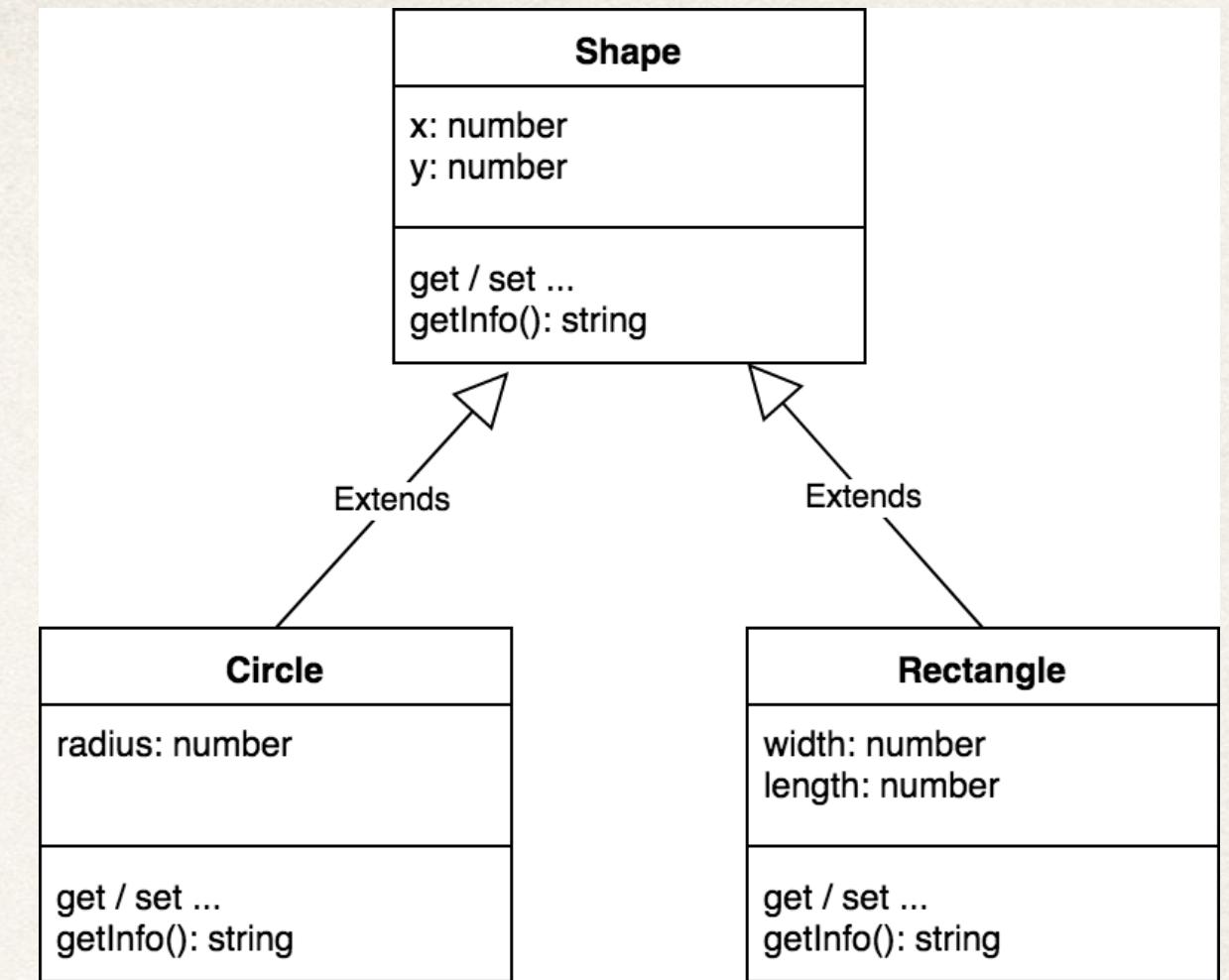
File: Driver.ts

```
import { Shape } from './Shape';
import { Circle } from './Circle';
import { Rectangle } from './Rectangle';

let myShape = new Shape(10, 15);
console.log(myShape.getInfo());

let myCircle = new Circle(5, 10, 20);
console.log(myCircle.getInfo());

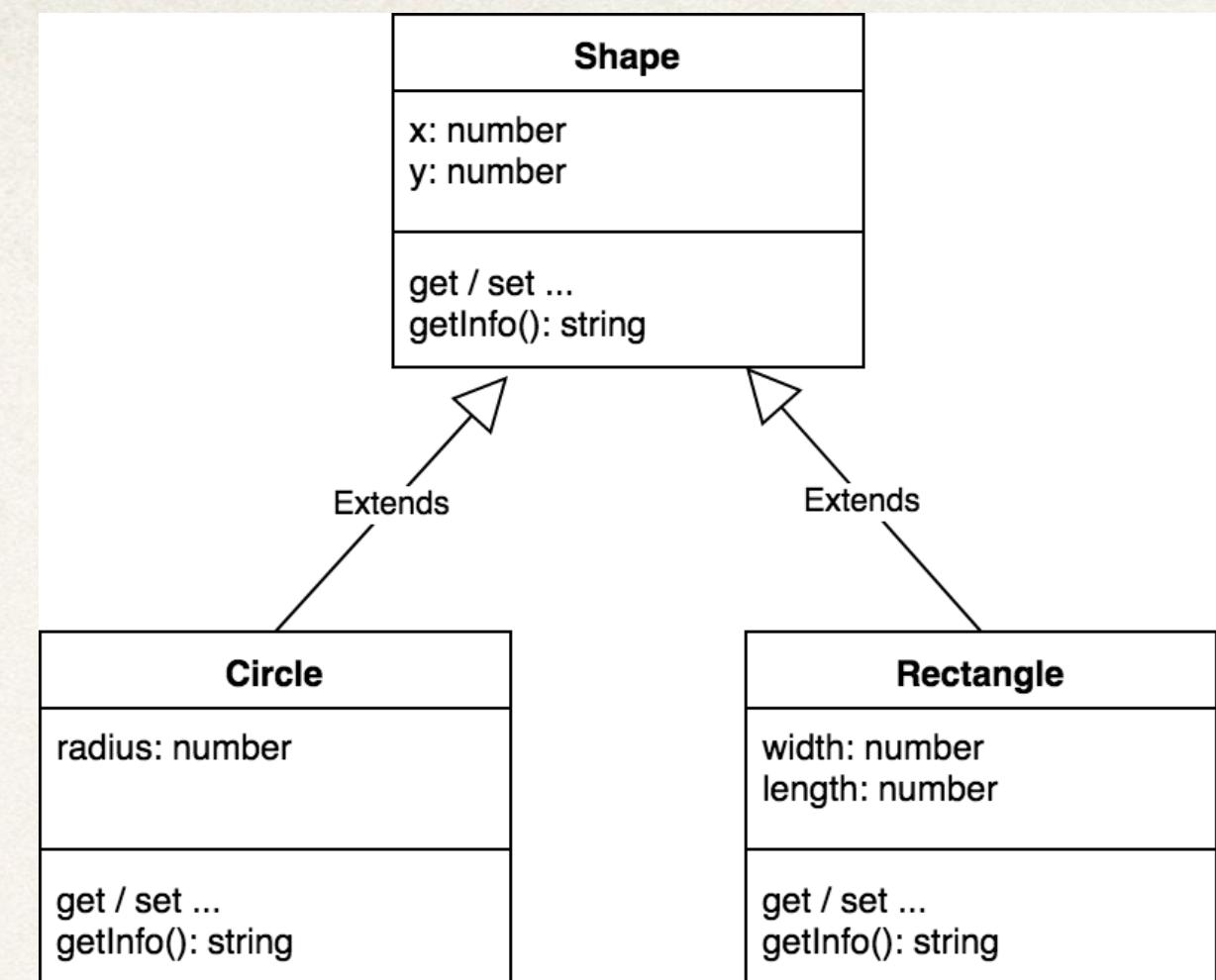
let myRectangle = new Rectangle(0, 0, 3, 7);
console.log(myRectangle.getInfo());
```



x=10, y=15  
x=5, y=10, radius=20  
x=0, y=0, width=3, length=7

# Creating an Array of Shapes

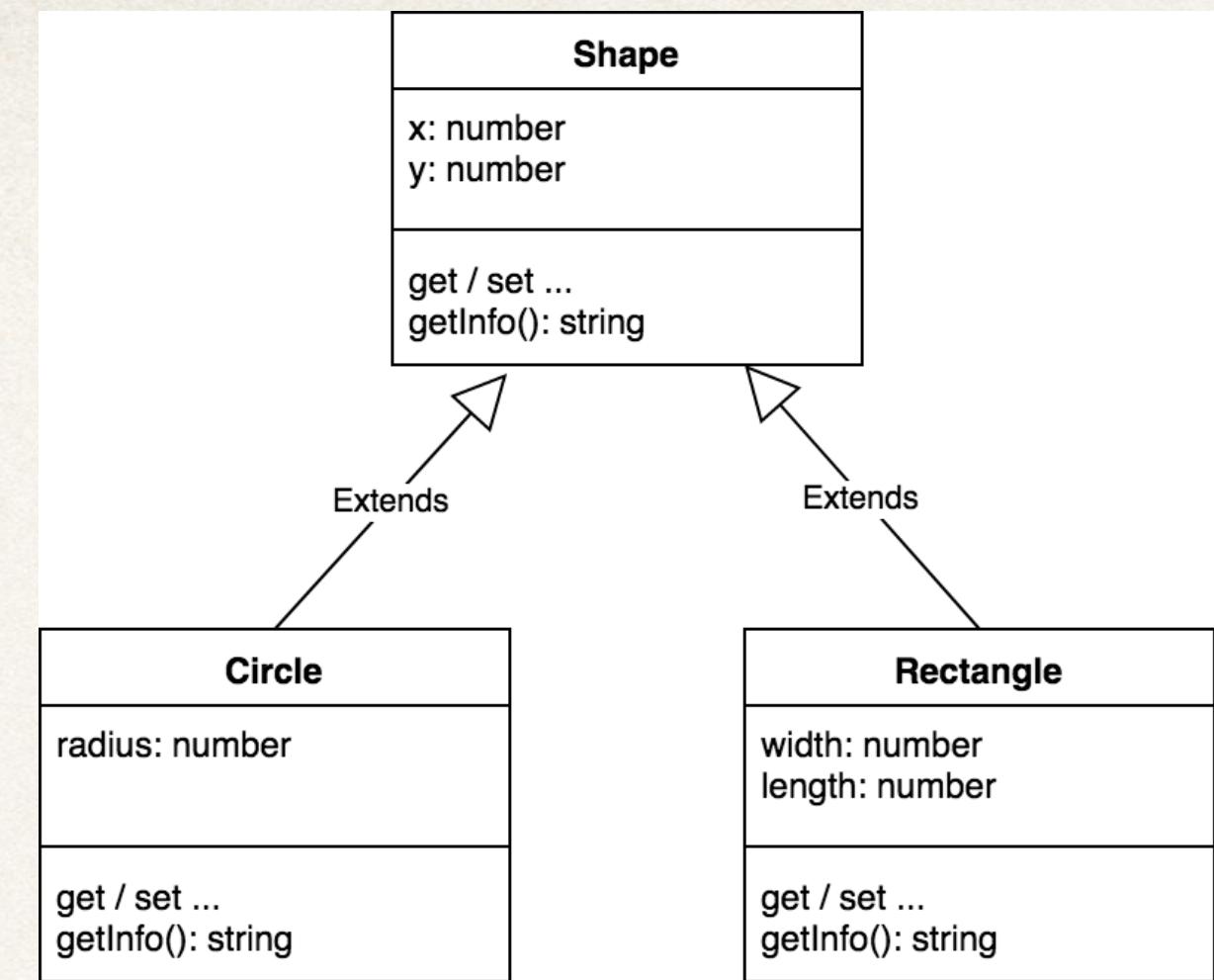
File: ArrayDriver.ts



# Creating an Array of Shapes

File: ArrayDriver.ts

```
...  
  
let myShape = new Shape(10, 15);  
let myCircle = new Circle(5, 10, 20);  
let myRectangle = new Rectangle(0, 0, 3, 7);
```



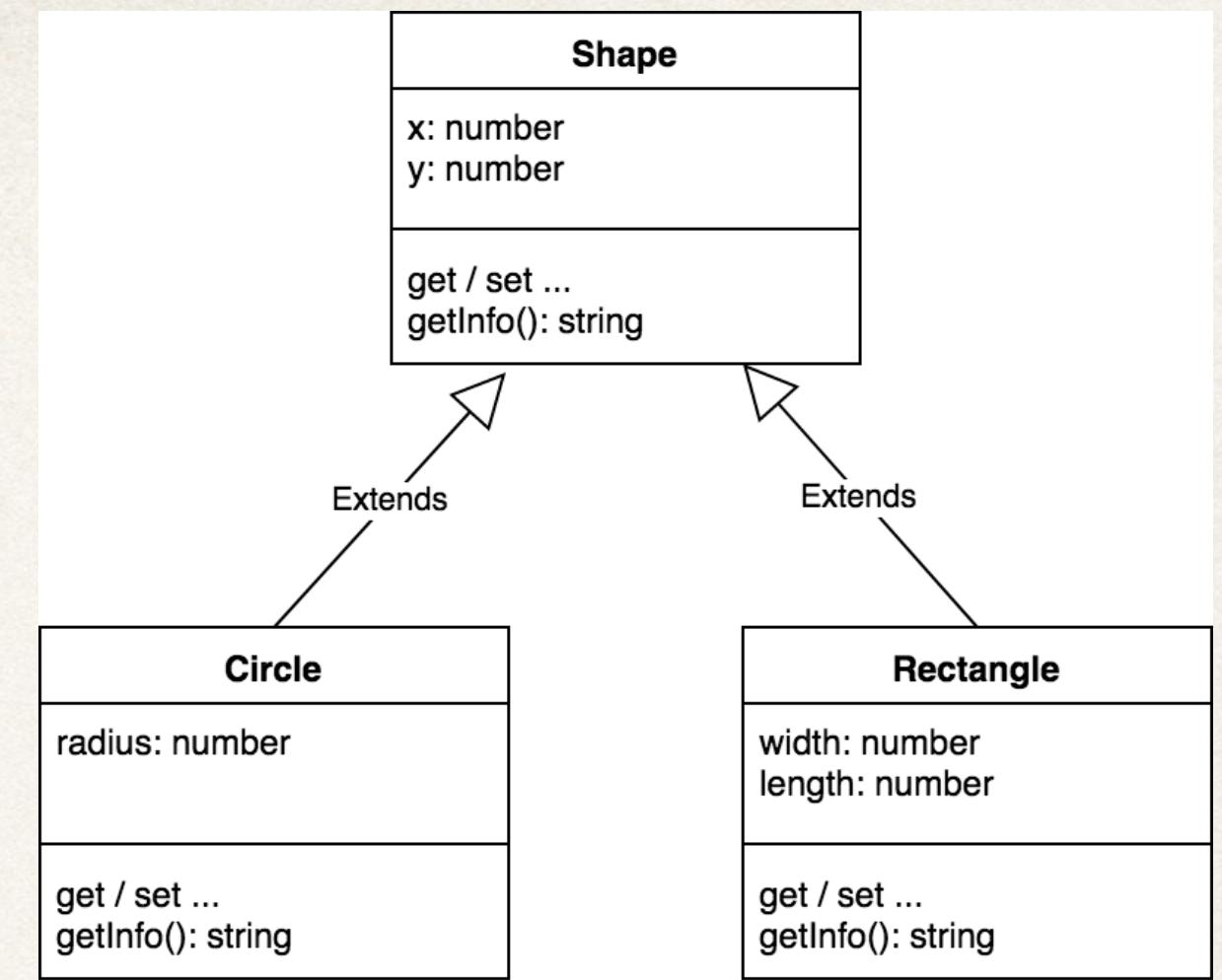
# Creating an Array of Shapes

File: ArrayDriver.ts

```
... ...

let myShape = new Shape(10, 15);
let myCircle = new Circle(5, 10, 20);
let myRectangle = new Rectangle(0, 0, 3, 7);

// declare an array for shapes ... initially empty
let theShapes: Shape[] = [];
```



# Creating an Array of Shapes

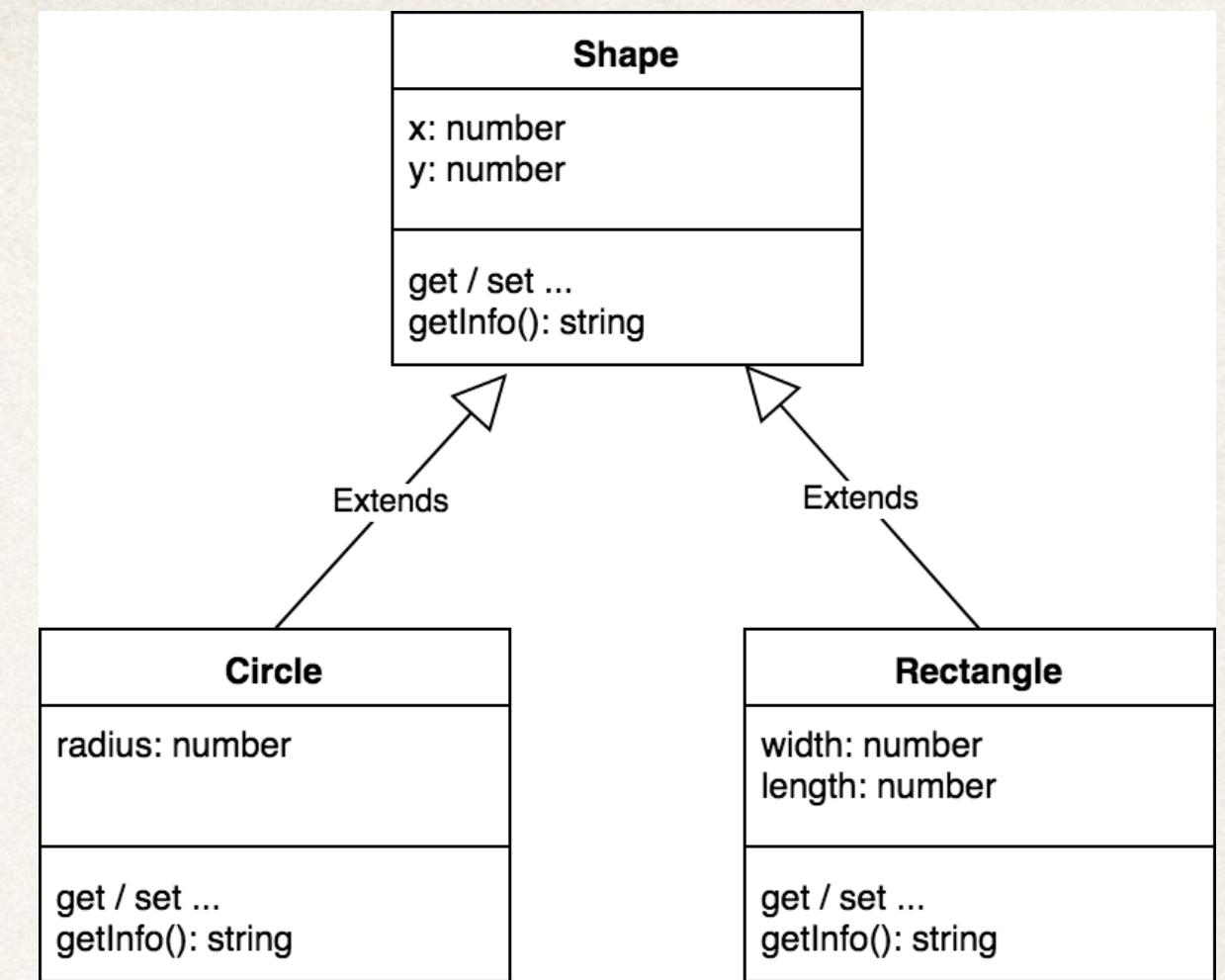
File: ArrayDriver.ts

```
... ...

let myShape = new Shape(10, 15);
let myCircle = new Circle(5, 10, 20);
let myRectangle = new Rectangle(0, 0, 3, 7);

// declare an array for shapes ... initially empty
let theShapes: Shape[] = [];

// add the shapes to the array
theShapes.push(myShape);
theShapes.push(myCircle);
theShapes.push(myRectangle);
```



# Creating an Array of Shapes

File: ArrayDriver.ts

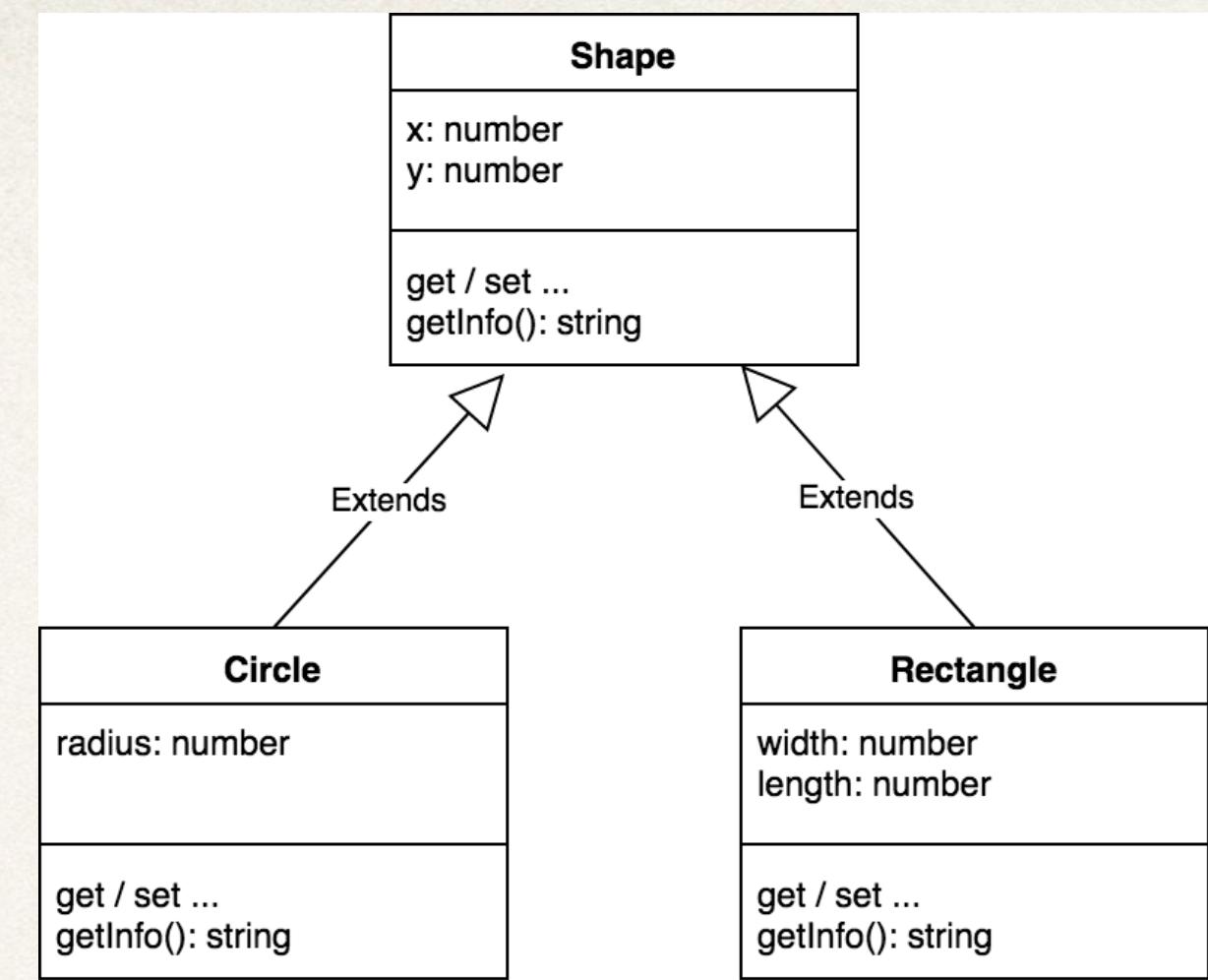
```
... ...

let myShape = new Shape(10, 15);
let myCircle = new Circle(5, 10, 20);
let myRectangle = new Rectangle(0, 0, 3, 7);

// declare an array for shapes ... initially empty
let theShapes: Shape[] = [];

// add the shapes to the array
theShapes.push(myShape);
theShapes.push(myCircle);
theShapes.push(myRectangle);

for (let tempShape of theShapes) {
    console.log(tempShape.getInfo());
}
```



# Creating an Array of Shapes

File: ArrayDriver.ts

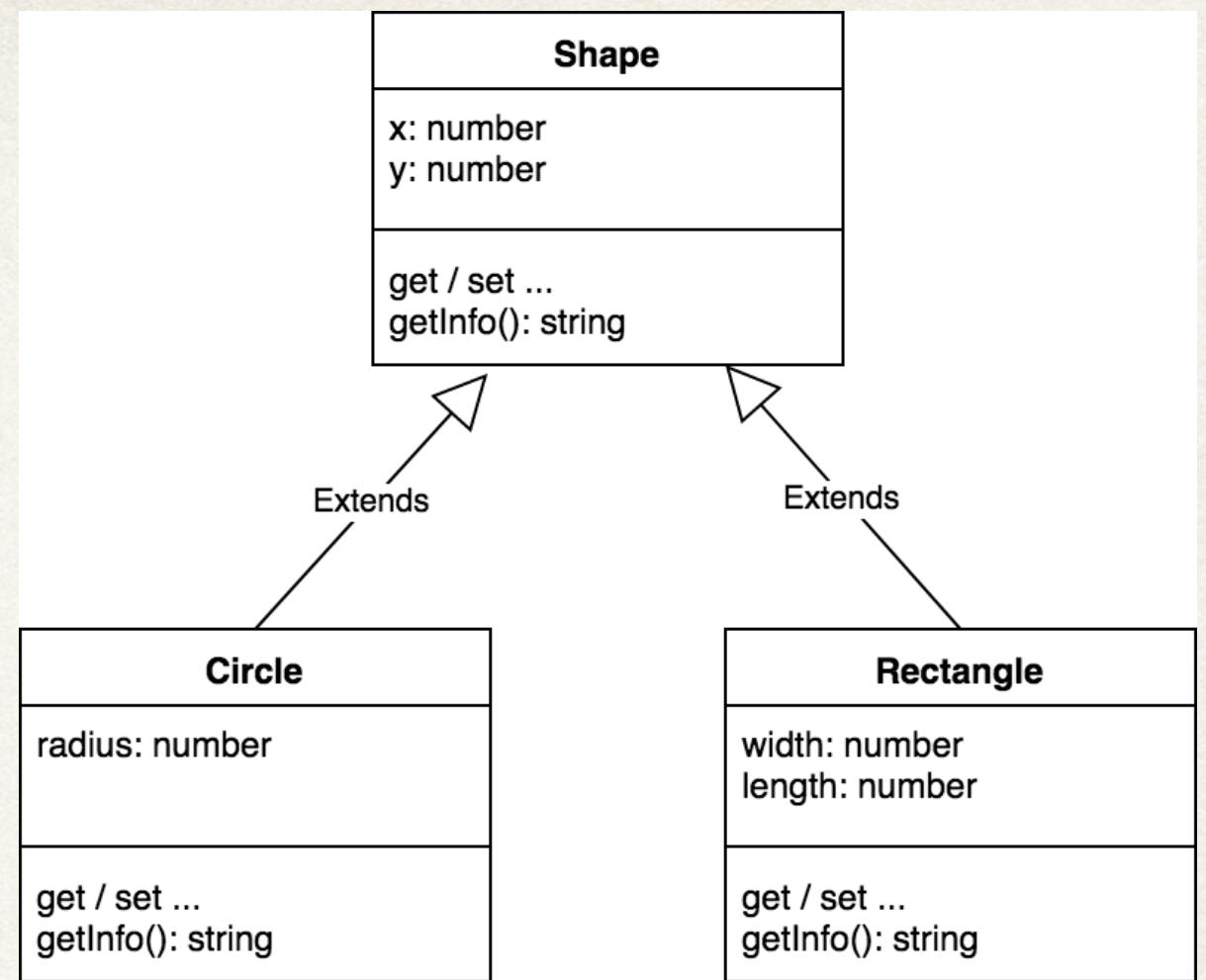
```
... ...

let myShape = new Shape(10, 15);
let myCircle = new Circle(5, 10, 20);
let myRectangle = new Rectangle(0, 0, 3, 7);

// declare an array for shapes ... initially empty
let theShapes: Shape[] = [];

// add the shapes to the array
theShapes.push(myShape);
theShapes.push(myCircle);
theShapes.push(myRectangle);

for (let tempShape of theShapes) {
    console.log(tempShape.getInfo());
}
```



x=10, y=15  
x=5, y=10, radius=20  
x=0, y=0, width=3, length=7

# Creating an Array of Shapes

File: ArrayDriver.ts

```
... ...

let myShape = new Shape(10, 15);
let myCircle = new Circle(5, 10, 20);
let myRectangle = new Rectangle(0, 0, 3, 7);

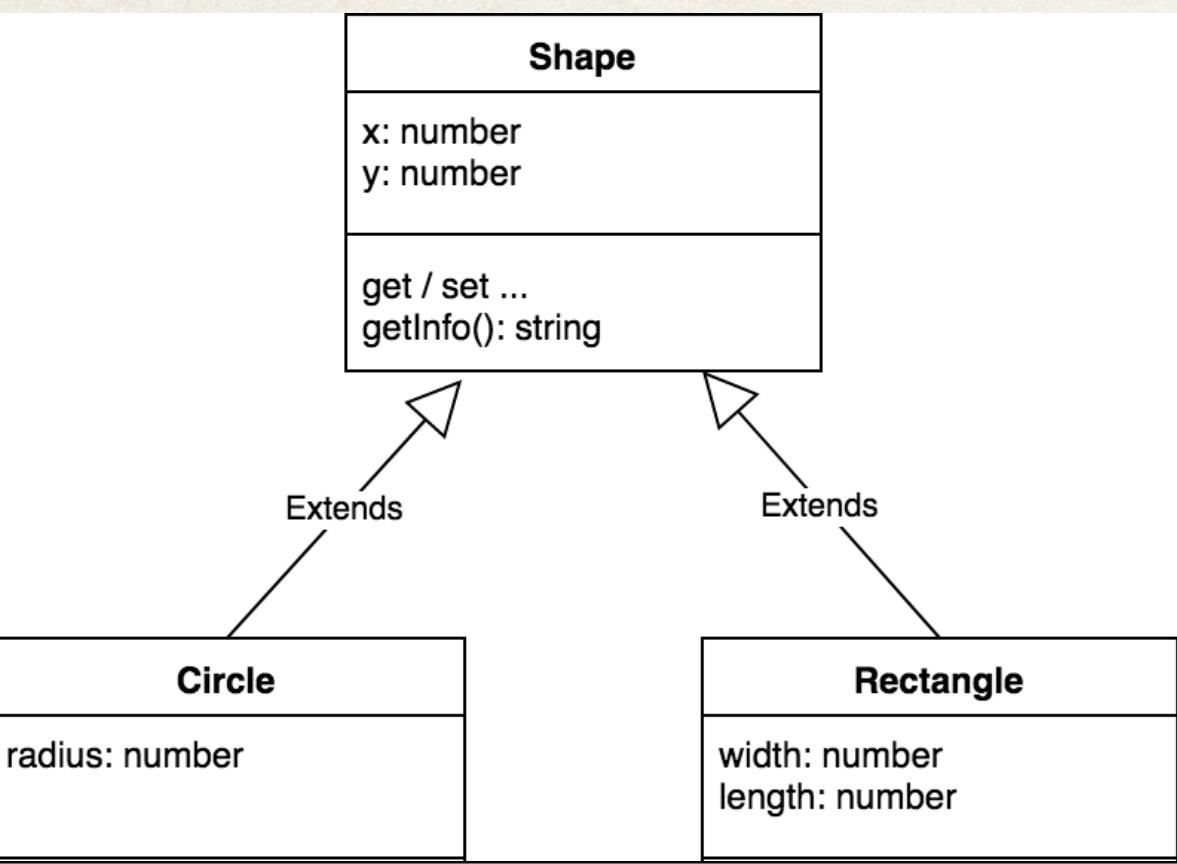
// declare an array for shapes ... initially empty
let theShapes: Shape[] = [];

// add the shapes to the array
theShapes.push(myShape);
theShapes.push(myCircle);
theShapes.push(myRectangle);

for (let tempShape of theShapes) {
    console.log(tempShape.getInfo());
}
```

```
// declare an array for shapes
let theShapes: Shape[] = [myShape, myCircle, myRectangle];
```

x=10, y=15  
x=5, y=10, radius=20  
x=0, y=0, width=3, length=7



# Creating an Array of Shapes

File: ArrayDriver.ts

```
...  
  
let myShape = new Shape(10, 15);  
let myCircle = new Circle(5, 10, 20);  
let myRectangle = new Rectangle(0, 0, 3, 7);  
  
// declare an array for shapes ... initially empty  
let theShapes: Shape[] = [];  
  
// add the shapes to the array  
theShapes.push(myShape);  
theShapes.push(myCircle);  
theShapes.push(myRectangle);  
  
for (let tempShape of theShapes) {  
    console.log(tempShape.getInfo());  
}
```

```
// declare an array for shapes  
let theShapes: Shape[] = [myShape, myCircle, myRectangle];
```

x=10, y=15  
x=5, y=10, radius=20  
x=0, y=0, width=3, length=7

