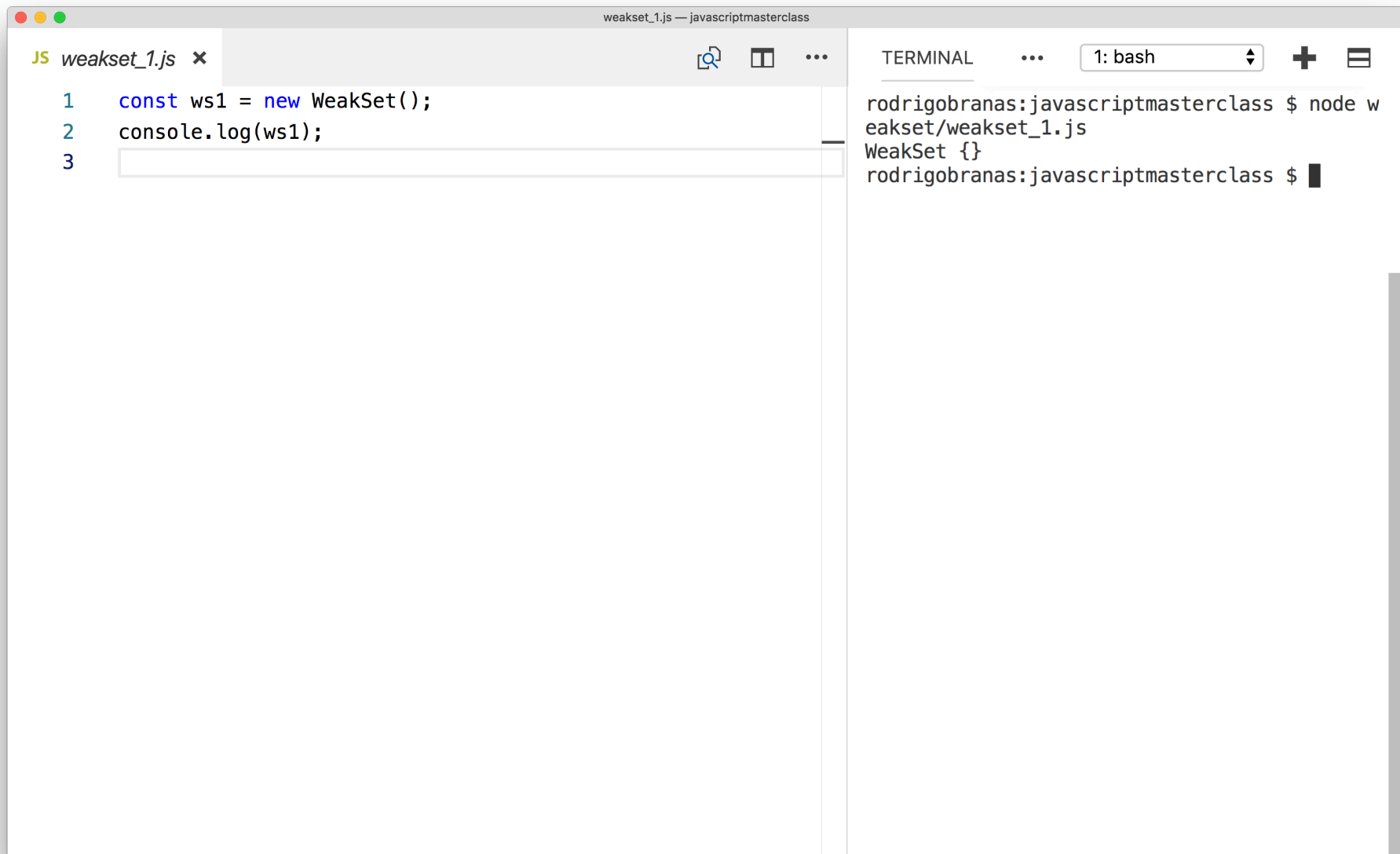


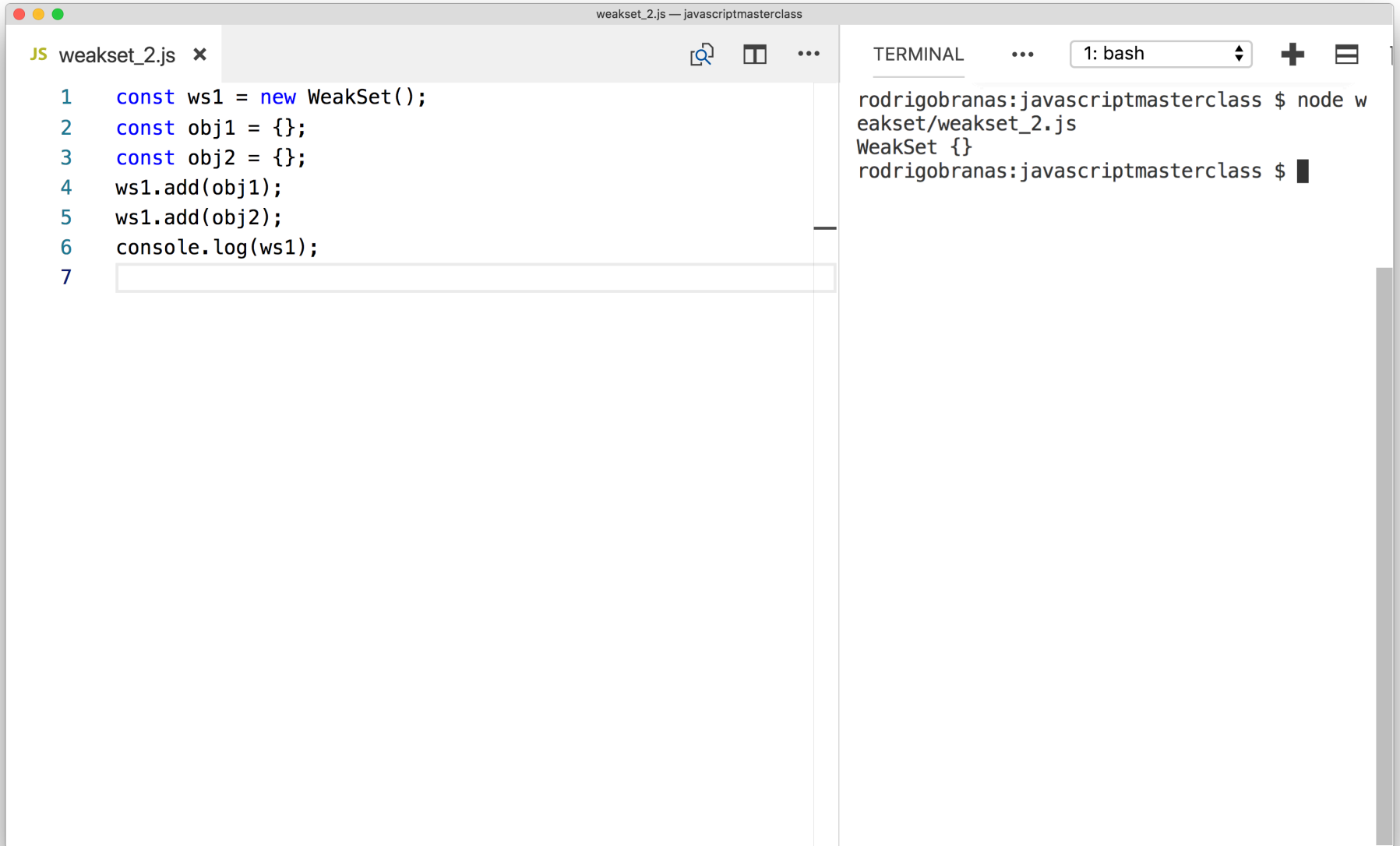


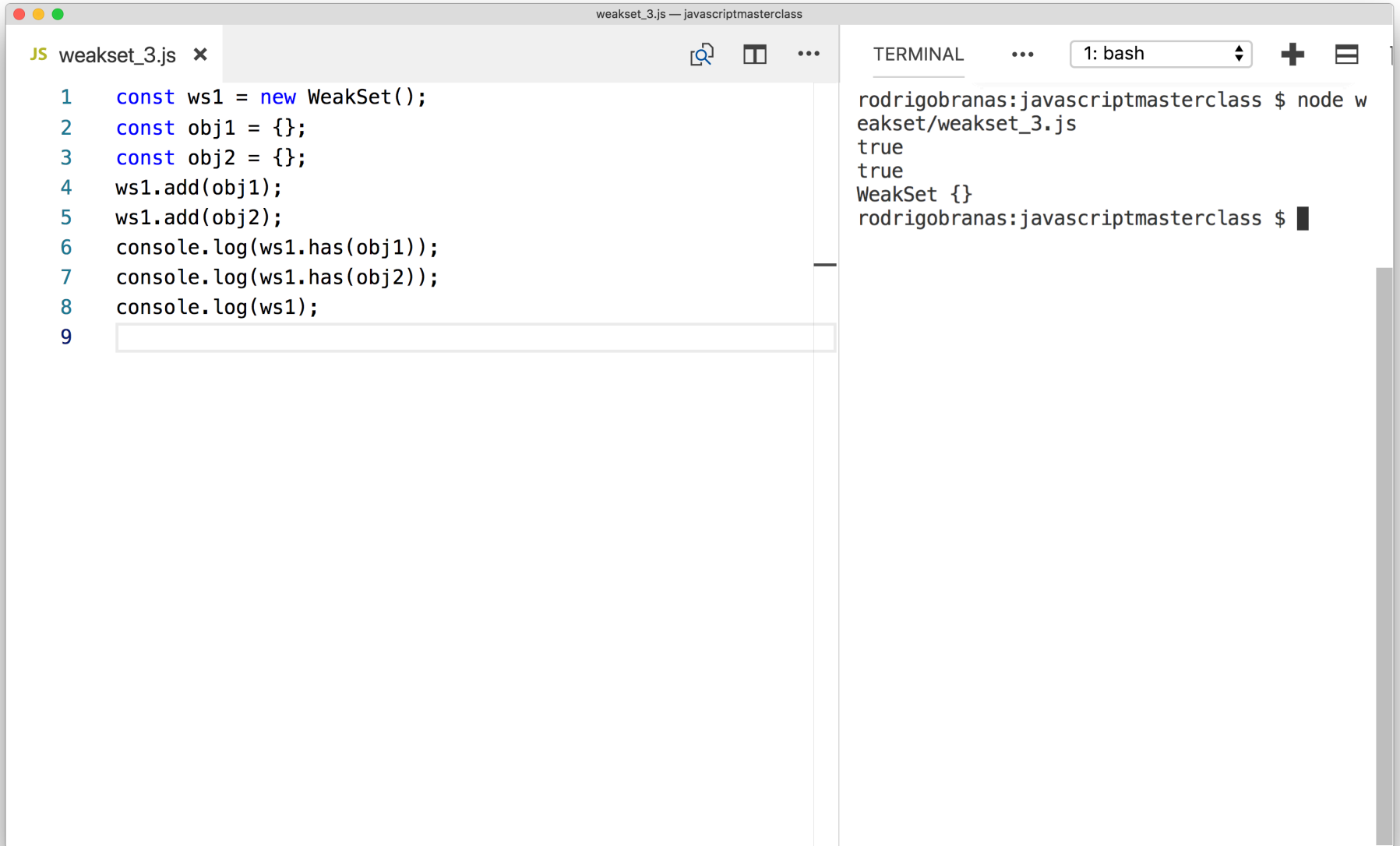
WeakSet

WeakSet é um objeto, similar ao Set, que **permite apenas valores do tipo Object** e mantém as referências de forma fraca, sendo volátil e não iterável



- **add**: Adiciona um elemento
- **has**: Retorna true se o elemento existir
- **delete**: Remove um elemento





JS weakset_4.js x



TERMINAL



1: bash



```
1  const ws1 = new WeakSet();
2  const obj1 = {};
3  const obj2 = {};
4  ws1.add(obj1);
5  ws1.add(obj2);
6  console.log(ws1.delete(obj2));
7  console.log(ws1.has(obj1));
8  console.log(ws1.has(obj2));
9  console.log(ws1);
10
```

```
rodrigobranas:javascriptmasterclass $ node weakset/weakset_4.js
true
true
false
WeakSet {}
rodrigobranas:javascriptmasterclass $
```



Pra que serve um **WeakSet**?

JS weakset_5.js x



TERMINAL



1: bash



```
1  const circles = new WeakSet();
2  function Circle(radius) {
3    circles.add(this);
4    this.radius = radius;
5  }
6  Circle.prototype.calculateArea = function() {
7    if (!circles.has(this)) throw "Invalid object";
8    return Math.PI * Math.pow(this.radius, 2);
9  };
10 const circle1 = new Circle(10);
11 const circle2 = {
12   radius: 5
13 };
14 console.log(circle1.calculateArea());
15 console.log(circle1.calculateArea.call(circle2));
16
```

```
rodrigobranas:javascriptmasterclass $ node weakset/weakset_5.js
314.1592653589793
```

```
/Users/rodrigobranas/development/workspace/javascriptmasterclass/weakset/weakset_5.js:7
    if (!circles.has(this)) throw "Invalid object";
                                ^
```

Invalid object

```
rodrigobranas:javascriptmasterclass $
```