Classes and Class Members

Classes, Constructors, Properties

SoftUni Team

Technical Trainers

Software University

SoftUni Foundation

Software University

http://softuni.bg

Table of Content

SoftUni Foundation

Have a Question?

SoftUni Foundation

sli.do

#JSCORE

Objects and Classes

Defining and Using Classes in JS

Objects

- In programming **objects** holds a set of named values
  - E.g. a **rectangle** object holds **width**, **height**

```
rect

width = 5
height = 4
color = 'red'
```

Object name

Object properties

```
▼ Object
    color: "red"
    height: 4
    width: 5
```

- Creating a "**rect**" object in JS:

```
let rect = { width: 5, height: 4, color: 'red' };
```

Classes

- In programming **classes** provide the **structure** for objects
  - Act as **template** for objects of the same type
- Classes define:
  - **Data** (properties, attributes), e.g. `width`, `height`, `color`
  - **Actions** (behavior), e.g. `calcArea()`, `resize(ratio)`
- One class may have many instances (objects)
- Example class: `Rectangle`
- Example objects: `redRect`, `blueRect`

Defining Classes in JS

SoftUni Foundation

```
class Rectangle {
  constructor(width, height, color) {
    this.width = width;
    this.height = height;
    this.color = color;
  }
}
let redRect = new Rectangle(4, 5, 'red');
let blueRect = new Rectangle(8, 3, 'blue');
console.log(redRect, blueRect);
```

The constructor defines class data

```
▶ Rectangle {width: 4, height: 5, color: "red"}
▶ Rectangle {width: 8, height: 3, color: "blue"}
```

```
▼ Rectangle i
    color: "red"
    height: 5
    width: 4
  ▶ __proto__: Object
```

```
▼ Rectangle i
    color: "blue"
    height: 3
    width: 8
  ▶ __proto__: Object
```

Create a new object

Classes Holding Data + Methods

SoftUni Foundation

```
class Rectangle {
  constructor(width, height, color) {
    [this.width, this.height, this.color] =
      [width, height, color];
  }
  calcArea() {
    return this.width * this.height;
  }
}
```

```
let rect = new Rectangle(4, 5, 'red');
console.log(rect.calcArea()); // 20
```

Methods perform operations over the class data

Check your solution here: https://judge.softuni.bg/Contests/336

Classes vs. Objects

SoftUni Foundation

Classes

```
class
Rectangle
```

Class name

```
width
height
color
```

Class data (properties)

```
calcArea()
resize(…)
```

Class actions (methods)

Objects

```
object
rect1
```

Object name

```
width = 4
height = 5
```

Object data

```
object
rect2
```

Object name

```
width = 8
height = 3
```

Object data

Software University

Problem: Persons

- Create a class **Person** to hold **firstName** + **lastName** + **age** + **email**
  - Define **`toString()`** method to print the person in this format:

```
{firstName} {lastName} (age: {age}, email: {email})
```

- Write a function **`getPersons()`** to return an array of the followin g persons:

First Name	Last Name	Age	Email
Maria	Petrova	22	mp@yahoo.com
SoftUni			
Stephan	Nikolov	25	
Peter	Kolev	24	ptr@gmail.com

Problem: Persons – Output

- The **getPersons()** function should work like this:

```
console.log(getPersons().join(", "));
```

```
Maria Petrova (age: 22, email: mp@yahoo.com),
SoftUni undefined (age: undefined, email: undefined),
Stephan Nikolov (age: 25, email: undefined),
Peter Kolev (age: 24, email: ptr@gmail.com)
```

Solution: Person Class

```
class Person {
  constructor(firstName, lastName, age, email) {
    [this.firstName, this.lastName, this.age, this.email] = [fi
rstName, lastName, age, email];
  }
  toString() {
    return `${this.firstName} ${this.lastName} (age: ${this.age
}, email: ${this.email})`;
  }
}
```

Check your solution here: https://judge.softuni.bg/Contests/336

Solution: getPersons() Function

```
function getPersons() {
  class Person { … }
  return [
    new Person('Maria', 'Petrova', 22, 'mp@yahoo.com'),
    new Person('SoftUni'),
    new Person('Stephan', 'Nikolov', 25),
    new Person('Peter', 'Kolev', 24, 'ptr@gmail.com'),
  ];
}
```

Check your solution here: https://judge.softuni.bg/Contests/336

```
▼ Circle {radius: 2}
    diameter: 4
    radius: 2
```

Accessor Properties

Defining Getters and Setters

Accessor Properties

SoftUni Foundation

```
class Circle {
  constructor(radius) { this.radius = radius; }
  get diameter() { return 2 * this.radius; }
  set diameter(diameter) {
    this.radius = diameter / 2;
  }
  get area() {
    return Math.PI * this.radius * this.radius;
  }
}
```

Property getter

Property setter

Read-only property "area"

Class Circle will hold property "radius" + accessor properties "diameter" and "area"

Accessor Properties in Action

SoftUni Foundation

```
let c = new Circle(2);
console.log(`Radius: ${c.radius}`); // 2
console.log(`Diameter: ${c.diameter}`); // 4
console.log(`Area: ${c.area}`); // 12.566370614359172
```

```
▼ Circle {radius: 2}
    area: 12.566370614359172
    diameter: 4
    radius: 2
  ▶ __proto__: Object
```

```
c.diameter = 1.6;
console.log(`Radius: ${c.radius}`); // 0.8
console.log(`Diameter: ${c.diameter}`); // 1.6
console.log(`Area: ${c.area}`); // 2.0106192982974678
```

```
▼ Circle {radius: 0.8}
    area: 2.0106192982974678
    diameter: 1.6
    radius: 0.8
```

static function (){}

Static Methods

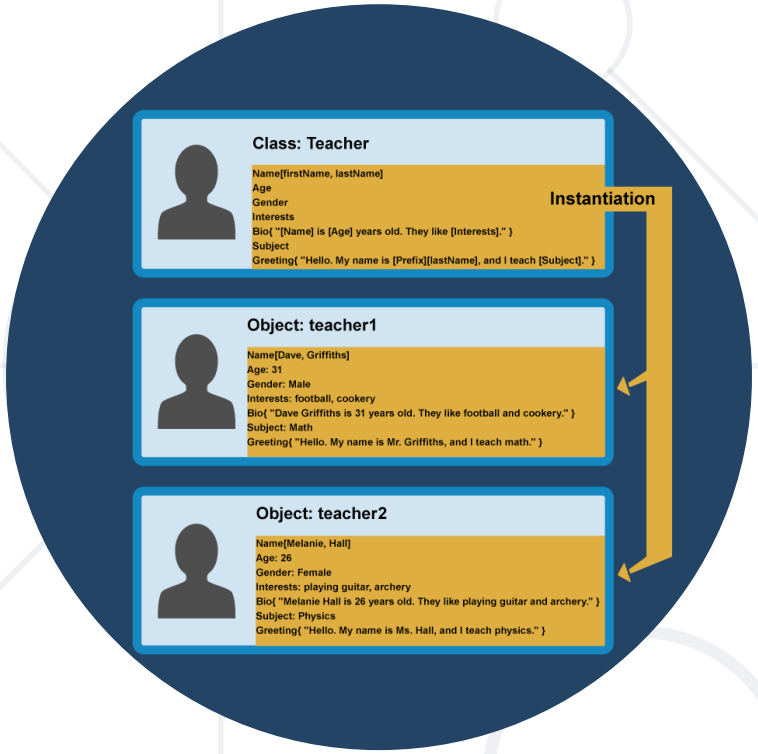
Static Methods

SoftUni Foundation

```
class Point {
  constructor(x, y) {
    this.x = x;
    this.y = y;
  }
  static distance(a, b) {
    const dx = a.x - b.x;
    const dy = a.y - b.y;
    return Math.sqrt(dx*dx + dy*dy);
  }
}
```

```
let p1 = new Point(5, 5);
let p2 = new Point(10, 10);
console.log(
  Point.distance(p1, p2));
```

Check your solution here: https://judge.softuni.bg/Contests/336



Legacy Classes in JS

Legacy Classes

SoftUni Foundation

- Before **ES2015** (ES6), classes were composed manually

```
function Rectangle(width, height) {
  this.width = width;
  this.height = height;
}
Rectangle.prototype.area = function () {
  return this.width * this.height;
}
let rect = new Rectangle(3, 5);
```

Constructor function defines class data

Behavior (methods) is later attached to the prototype

Instantiation works the same way

Check your solution here: https://judge.softuni.bg/Contests/336

Comparison with the New Syntax

SoftUni Foundation

```
class Rectangle {
  constructor(width, height) {
    this.width = width;
    this.height = height;
  }

  area() {
    return this.width * this.height;
  }
}
```

```
function Rectangle(width, height)  {
  this.width = width;
  this.height = height; }
```

```
Rectangle.prototype.area = function() {
  return this.width * this.height;}
```

Protecting Class Data

Keeping the Class State Correct

Read Only Class Data

SoftUni Foundation

```
class Cat {
  constructor(name, age) {
    this.name = name;
    this.age = age;
    Object.freeze(this);
  }
}
```


```
let c = new Cat('Garfield',  5);
c.name = 'Tom'; // not working (Error in strict mode)
c.sex = 'M';    // not working (Error in strict mode)
console.log(c); // Cat { name: 'Garfield', age: 5 }
```

Inextensible Class Data

SoftUni Foundation

```
class Cat {
  constructor(name, age) {
    this.name = name;
    this.age = age;
    Object.seal(this);
  }
}
```

```
let c = new Cat('Garfield',  5);
c.name = 'Tom'; // OK
c.sex = 'M';     // not working (Error in strict mode)
console.log(c); // Cat { name: 'Tom', age: 5 }
```

Problem: Cards

- Write a function that returns a class **`Card`** and a enum **`Suits`**:
  - **`Suits`** is an object with keys [**`SPADES`**, **`HEARTS`**, **`DIAMONDS`**, **`CLUBS`**] and corresponding values [**♠, ♥, ♦, ♣**]
  - The **`Card`** class should hold a **Face** + **Suit**:
    - **Face** must be in [**2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A**]
    - **Suit** must be a value from **Suites**
  - **`Card.toString()`** should return the card as text, e.g. **K♦**
  - Creating an invalid Card (e.g. **-1♥**) should throw an Error

Problem: Cards – Sample Output

```
let defineCards = (function() { … } () )
let Suits = defineCards.Suites;
let Card = defineCards.Card;
```

```
let card = new Card("Q",Suits.DIAMONDS));
console.log('' + card); // Q♦
```

→ OK

```
let card = new Card("1",Suits.DIAMONDS);
```

→ Error

```
let card = new Card("A",Suits.Pesho);
```

→ Error

```
let card = new Card("A",'hearts');
```

→ Error

Solution: Create Cards Function

```
(function() {
  let Suits = {
    CLUBS: "\u2663",     // ♣
    DIAMONDS: "\u2666", // ♦
    HEARTS: "\u2665",    // ♥
    SPADES: "\u2660"     // ♠
  };
  let Faces = ['2', '3', '4', '5', '6', '7', '8', '9', '10',
'J', 'Q', 'K', 'A'];
  class Card { … }
  return { Suits, Card }
}())
```

Solution: Class Card

```
class Card {
  constructor(face, suit) {
    this.suit = suit;
    this.face = face;
  }

  get face() { return this._face; }
  set face(face) {
    if (!Faces.includes(face))
      throw new Error("Invalid card face: " + face);
    this._face = face;
  }
  …
```

Use different identifier **this._face** to avoid infinite recursion: **set suit()** invoking itself

Solution: Class Card (2)

```
class Card {

  …
  get suit() { return this._suit; }
  set suit(suit) {
    if (!Object.keys(Suits).map(
        k => Suits[k]).includes(suit))
      throw new Error("Invalid card suite: " + suit);
    this._suit = suit;
  }

  toString() { return `${this.face}${this.suit}`; }

}
```

Check your solution here: https://judge.softuni.bg/Contests/336

Live Exercises

Summary

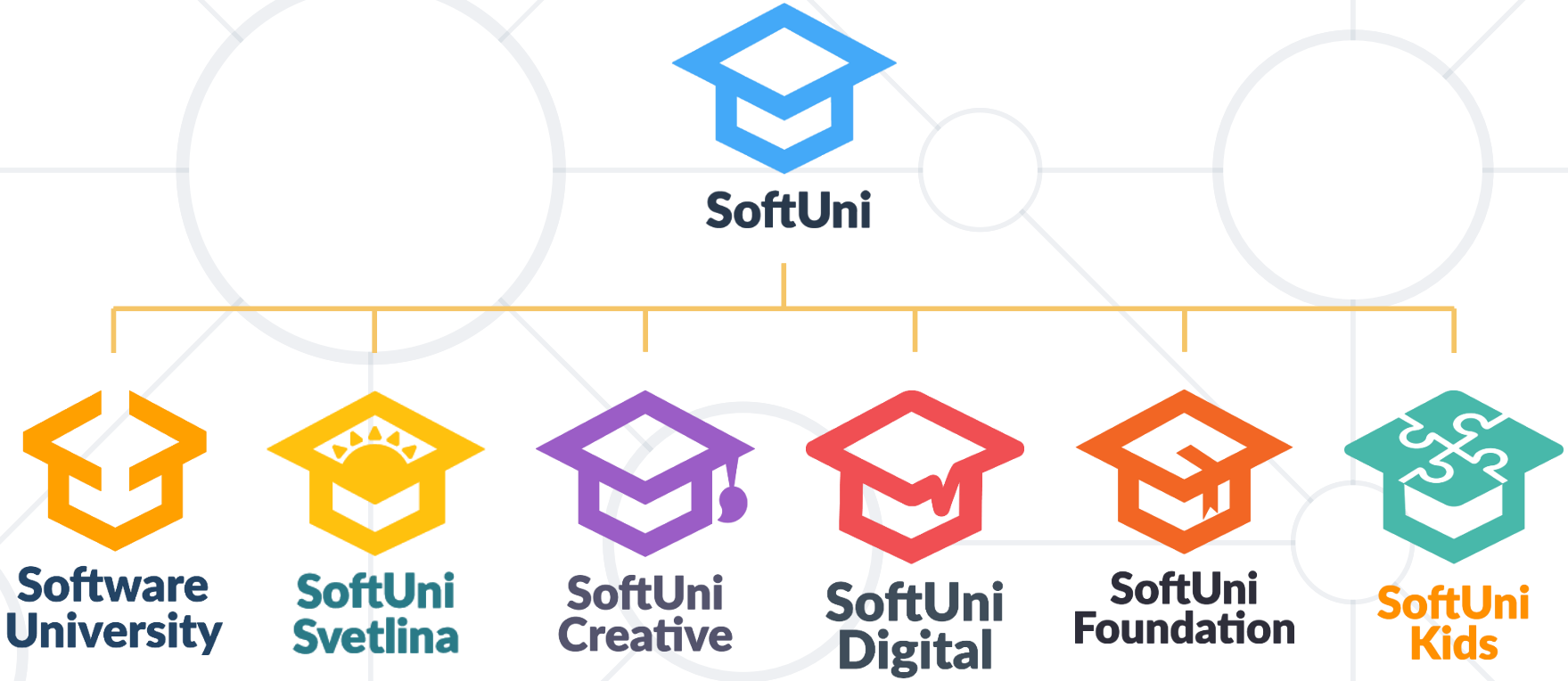
- Classes provide structure for objects
- Classes may define methods:

```
toString() { … }
```

- Classes may define accessor properties
  - Getters and setters:

```
get area() { return … }
```

Questions?



https://softuni.bg/trainings/2081/js-advanced-october-2018

SoftUni Diamond Partners

SoftUni Foundation

XSsoftware

SBTech we know sports

telenor

SoftwareGroup doing it right

NETPEAK

SmartIT

Postbank Решения за твоето утре

SUPER HOSTING .BG

INDEAVR Serving the high achievers

INFRAGISTICS

LIEBHERR

æternity

codexio

SoftUni Organizational Partners

SoftUni Foundation

ИНФОРМАЦИОННО ОБСЛУЖВАНЕ

OneBit SOFTWARE

WORLD OF MYTHS

Lukanet.com

codexio

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education and Employment Opportunities
  - softuni.bg
- Software University Foundation
  - http://softuni.foundation/
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license