



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Task № 2

Дисциплина: Архитектура ЭВМ

Тема : Основы ООП.

Студент Расколотов Д.Ю.

Группа ИУ7-44Б

Преподаватель Повов А.Ю.

Москва.
2020 г.

Цель работы: Познакомиться с ООП в NodeJS. А так же изучить использование setTimeout и setInterval.

Задание 1:

Создать класс *Точка*.

Добавить классу *Точка* метод инициализации полей и метод вывода полей на экран

Создать класс *Отрезок*.

У класса *Отрезок* должны быть поля, являющиеся экземплярами класса *Точка*.

Добавить классу *Отрезок* метод инициализации полей, метод вывода информации о полях на экран, а так же метод получения длины отрезка.

Код программы:

```
"use strict"

class Dot{
  constructor(x, y) {
    this.x = x
    this.y = y
  }

  get output(){
    console.log(`x = ${this.x} y = ${this.y}`)
  }
}

class Section{
  constructor(dot1, dot2) {
    this.dot1 = dot1
    this.dot2 = dot2
  }
}
```

```

    get output() {
        console.log(`st_x = ${this.dot1.x} st_y = ${this.dot1.y}
        \nend_x = ${this.dot2.x} end_y = ${this.dot2.y}`)
    }

    get dlina(){
        return Math.sqrt(Math.pow(this.dot1.x - this.dot2.x, 2)
        + Math.pow(this.dot1.y - this.dot2.y, 2))
    }
}

function main() {
    let dot1 = new Dot(100, 200)
    let dot2 = new Dot(300, 400)

    dot1.output
    dot2.output

    let s = new Section(dot1, dot2)
    console.log(s.dlina)
    s.output
}

main()

```

Тестовая часть:

```

x = 100 y = 200
x = 300 y = 400
282.842712474619
st_x = 100 st_y = 200
end_x = 300 end_y = 400

```

Задание 2:

Создать класс *Треугольник*.

Класс *Треугольник* должен иметь поля, хранящие длины сторон треугольника.

Реализовать следующие методы:

- Метод инициализации полей
- Метод проверки возможности существования треугольника с такими сторонами
- Метод получения периметра треугольника
- Метод получения площади треугольника
- Метод для проверки факта: является ли треугольник прямоугольным

Код программы:

```
"use strict"

class Triangle {
  constructor(a, b, c) {
    this.a = a
    this.b = b
    this.c = c
  }

  get exist_triangle() {
    if (this.a + this.b > this.c && this.a + this.c > this.b && this.b + this.c > this.a)
      return true
    else
      return false
  }

  get perimetr() {
    if (this.exist_triangle)
      return this.a + this.b + this.c
    else
      return NaN
  }
}
```

```

    get square(){
        if (this.exist_triangle) {
            let p = this.perimetr / 2
            return Math.sqrt(p * (p - this.a) * (p - this.b) *
(p - this.c))
        }
        else
            return NaN
    }

    get is_rectangular(){
        if (this.exist_triangle)
            return Math.abs(Math.pow(this.a,2) + Math.pow(this.
b,2) - Math.pow(this.c,2)) < 0.01
        else
            return NaN
    }
}

function main(){
    let tr = new Triangle(4,4,5)

    console.log(tr.exist_triangle)

    console.log(tr.perimetr)

    console.log(tr.square)

    console.log(tr.is_rectangular)

}

main()

```

Тестовая часть:

```
true
13
7.806247497997997
false
```

Задание 3:

Реализовать программу, в которой происходят следующие действия:

Происходит вывод целых чисел от 1 до 10 с задержками в 2 секунды.

После этого происходит вывод от 11 до 20 с задержками в 1 секунду.

Потом опять происходит вывод чисел от 1 до 10 с задержками в 2 секунды.

После этого происходит вывод от 11 до 20 с задержками в 1 секунду.

Это должно происходить циклически.

Код программы:

```
"use strict";

let num = 0;
let flag = 0;

let first_interval = setInterval(() => {
  if (!flag) {
    num++;
    console.log(num);
    if (num === 10) {
      flag = 1;
    }
  }
}, 2000);

let second_interval = setInterval(() => {
  if (flag) {
    num++;
  }
}, 1000);
```

```
    console.log(num);  
    if (num === 20) {  
        flag = 0;  
        num = 0;  
    }  
}  
}, 1000);
```

Тестовая часть:

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
1  
2
```

Заключительная часть:

Изучил тему: «Основы ООП». Разобрался с работой setInterval и setTimeout.