Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет информатики и радиоэлектроники» Филиал  
«Минский радиотехнический колледж»

Учебная дисциплина «Программные средства создания Internet-приложений»

**Отчет**

по выполнению лабораторной работы

«Применение объектно-ориентированного подхода в программировании на языке JavaScript»

Выполнила Гончаров П.В.

Проверила Терешко О.И.

Минск 2020

Лабораторная работа № 27

Номер учебной группы: 7K2492.

Фамилия, инициалы учащегося: Гончаров П.В.

Дата выполнения работы: 03.03.2021

Тема работы: «Применение объектно-ориентированного подхода в программировании на языке JavaScript»

Цель работы: Формирование умений применения объектно-ориентированного подхода в программировании на языке JavaScript.

Результат выполнения работы:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Lab 27</title>

    <script src="script.js"></script>

</head>

<body>

</body>

</html>

let zooPets = [];

let countZoopet= 0;

let countMammal = 0;

let countBird = 0;

class ZooPet {

  constructor(name, age, sound, children, isMale) {

    this.name = name;

    this.age = age;

    this.sound = sound;

    this.children = children;

    this.isMale = isMale;

    countZoopet++;

    zooPets.push(this);

  }

  showZooPetInfo() {

    console.log(`${this.name}(${this.age}): ${this.sound}`)

  }

  static getNumberZooPetsByType() {

    const numberZooPetsByType = {};

    zooPets.forEach(zooPet => {

      const existingNumberZooPetByType = numberZooPetsByType[zooPet.constructor.name];

      if(existingNumberZooPetByType !== undefined) {

        numberZooPetsByType[zooPet.constructor.name] = existingNumberZooPetByType + 1;

      }

      else{

        numberZooPetsByType[zooPet.constructor.name] = 1;

      }

    });

    return numberZooPetsByType;

  }

}

class Mammal extends ZooPet {

  constructor(

    name, age, sound, children, isMale,

    woolColor

  ) {

    super(name, age, sound, children, isMale);

    this.woolColor = woolColor;

    countMammal++;

  }

}

class Bird extends ZooPet {

  constructor(name, age, sound, children, isMale,

    feathersColor

  ) {

    super(name, age, sound, children, isMale);

    this.feathersColor = feathersColor;

    countBird++;

  }

}

const seaUrchin = new ZooPet("Ежик", 20, "иии");

const childLion = new Mammal("Симба", 1, "рь-pь-рь", [], true, "оранжевый");

const dadLion = new Mammal("Муфаса", 8, "рpр-ррр", [childLion], true, "оранжевый");

const mumLion = new Mammal("Сараби", 8, "рpр", [childLion], false, "оранжевый");

const penguin1 = new Bird("Шкипер", 7, "Улыбаемся и машем, парни.", [], true, "черно-белый");

const penguin2 = new Bird("Ковальски", 7, "Не стоит недооценивать простую силу пингвиньей неотразимости.", [], true, "черно-белый");

const penguin3 = new Bird("Рико", 7, "Кабум! Кабум! Кабум!", [], true, "черно-белый");

let penguin4 = new Bird("Шкипер", 5, "Меня запеченькали!", [], true, "черно-белый");

penguin1.showZooPetInfo();

console.log(zooPets);

console.log(`countZoopet:${countZoopet - countMammal - countBird}, countMammal:${countMammal}, countBird:${countBird}`);

console.log(ZooPet.getNumberZooPetsByType());



Контрольные вопросы и задания

1. В чем суть прототипного наследования?

Когда один класс расширяет другой

2. Перечислите известные Вам способы создания объектов в JavaScript?

Функция-конструктор, класс, связывание объектов, фабричная функция.

3. Приведите пример наследования с использованием классов.

class Mammal extends ZooPet {

  constructor(

    name, age, sound, children, isMale,

    woolColor

  ) {

    super(name, age, sound, children, isMale);

    this.woolColor = woolColor;

    countMammal++;

  }

}