МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ассистент |  |  |  | Ю.В. Ветрова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ПРАКТИЧЕСКОМ ЗАДАНИИ №3 |
|  |
| по курсу: JavaScript, его библиотеки и фреймворки в Frontend-разработке |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4136 |  |  |  | Бобрович Н.С. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2024

1. **Цель работы:**

формирование практических навыков работы с прототипами и классами в JavaScript.

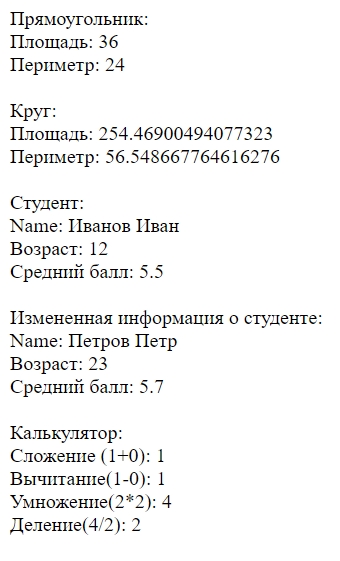
1. **Формулировка заданий:**

1. Создайте прототип "Фигура", который будет иметь методы для вычисления площади и периметра фигуры. Напишите классы "Прямоугольник" и "Круг", которые наследуются от прототипа "Фигура" и реализуют соответствующие методы.

2. Создайте класс "Студент", который будет иметь защищенные свойства "имя", "возраст" и "средний балл". Напишите методы для изменения и получения значений этих свойств.

3. Создайте класс "Калькулятор", который будет иметь методы для выполнения основных математических операций (сложение, вычитание, умножение, деление).

1. **Скриншоты, иллюстрирующие выполнение отдельных пунктов задания:**



1. **Листинг:**

Lab3.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>lab3</title>

</head>

<body>

<span id="outputSpan"></span>

<script src="lab3.js"></script>

</body>

</html>

Lab3.ts:

class Figure {

calcArea() {

throw new Error("Метод вычисления площади должен быть переопределен");

}

calcPerimeter() {

throw new Error("Метод вычисления периметра должен быть переопределен");

}

}

class Rectangle extends Figure {

public width:number;

public height:number;

constructor(width: number, height: number) {

super();

this.width = width;

this.height = height;

}

calcArea() {

return this.width \* this.height;

}

calcPerimeter() {

return 2 \* (this.width + this.height);

}

}

class Circle extends Figure {

public radius:number;

constructor(radius: number) {

super();

this.radius = radius;

}

calcArea() {

return Math.PI \* this.radius \* this.radius;

}

calcPerimeter() {

return 2 \* Math.PI \* this.radius;

}

}

class Student {

protected studentName: string;

protected age: number;

protected grades: number;

constructor(studentName: string, age: number, grades: number) {

this.studentName = studentName;

this.age = age;

this.grades = grades;

}

getName() {

return this.studentName;

}

getAge() {

return this.age;

}

getGrades() {

return this.grades;

}

setName(newName: string) {

this.studentName = newName;

}

setAge(newAge: number) {

this.age = newAge;

}

setGrades(newGrades: number) {

this.grades = newGrades;

}

}

class Calc {

Addition(a: any, b: any) {

return a + b;

}

Subtraction(a: number, b: number) {

return a - b;

}

Multiply(a: number, b: number) {

return a \* b;

}

Division(a: number, b: number) {

return a / b;

}

}

let rect = new Rectangle(6, 6);

let circ = new Circle(9);

let student = new Student('Иванов Иван', 12, 5.5);

let calc = new Calc();

let output = document.getElementById('outputSpan');

output.innerHTML += `Прямоугольник: <br> Площадь: ${rect.calcArea()} <br> Периметр: ${rect.calcPerimeter()} <br><br>`;

output.innerHTML += `Круг: <br> Площадь: ${circ.calcArea()} <br> Периметр: ${circ.calcPerimeter()} <br><br>`;

output.innerHTML += `Студент: <br> Name: ${student.getName()} <br> Возраст: ${student.getAge()} <br> Средний балл: ${student.getGrades()} <br><br>`;

student.setName('Петров Петр');

student.setAge(23);

student.setGrades(5.7);

output.innerHTML += `Измененная информация о студенте: <br> Name: ${student.getName()} <br> Возраст: ${student.getAge()} <br> Средний балл: ${student.getGrades()} <br><br>`;

output.innerHTML += `Калькулятор: <br> Сложение (0+1): ${calc.Addition(0, 1)} <br> Вычитание(1-0): ${calc.Subtraction(1, 0)} <br> Умножение(2\*2): ${calc.Multiply(2, 2)} <br> Деление(4/2): ${calc.Division(4, 2)} <br><br>`;

Lab3.js: (.ts -> .js)

var \_\_extends = (this && this.\_\_extends) || (function () {

var extendStatics = function (d, b) {

extendStatics = Object.setPrototypeOf ||

({ \_\_proto\_\_: [] } instanceof Array && function (d, b) { d.\_\_proto\_\_ = b; }) ||

function (d, b) { for (var p in b) if (Object.prototype.hasOwnProperty.call(b, p)) d[p] = b[p]; };

return extendStatics(d, b);

};

return function (d, b) {

if (typeof b !== "function" && b !== null)

throw new TypeError("Class extends value " + String(b) + " is not a constructor or null");

extendStatics(d, b);

function \_\_() { this.constructor = d; }

d.prototype = b === null ? Object.create(b) : (\_\_.prototype = b.prototype, new \_\_());

};

})();

var Figure = /\*\* @class \*/ (function () {

function Figure() {

}

Figure.prototype.calcArea = function () {

throw new Error("Метод вычисления площади должен быть переопределен");

};

Figure.prototype.calcPerimeter = function () {

throw new Error("Метод вычисления периметра должен быть переопределен");

};

return Figure;

}());

var Rectangle = /\*\* @class \*/ (function (\_super) {

\_\_extends(Rectangle, \_super);

function Rectangle(width, height) {

var \_this = \_super.call(this) || this;

\_this.width = width;

\_this.height = height;

return \_this;

}

Rectangle.prototype.calcArea = function () {

return this.width \* this.height;

};

Rectangle.prototype.calcPerimeter = function () {

return 2 \* (this.width + this.height);

};

return Rectangle;

}(Figure));

var Circle = /\*\* @class \*/ (function (\_super) {

\_\_extends(Circle, \_super);

function Circle(radius) {

var \_this = \_super.call(this) || this;

\_this.radius = radius;

return \_this;

}

Circle.prototype.calcArea = function () {

return Math.PI \* this.radius \* this.radius;

};

Circle.prototype.calcPerimeter = function () {

return 2 \* Math.PI \* this.radius;

};

return Circle;

}(Figure));

var Student = /\*\* @class \*/ (function () {

function Student(studentName, age, grades) {

this.studentName = studentName;

this.age = age;

this.grades = grades;

}

Student.prototype.getName = function () {

return this.studentName;

};

Student.prototype.getAge = function () {

return this.age;

};

Student.prototype.getGrades = function () {

return this.grades;

};

Student.prototype.setName = function (newName) {

this.studentName = newName;

};

Student.prototype.setAge = function (newAge) {

this.age = newAge;

};

Student.prototype.setGrades = function (newGrades) {

this.grades = newGrades;

};

return Student;

}());

var Calc = /\*\* @class \*/ (function () {

function Calc() {

}

Calc.prototype.Addition = function (a, b) {

return a + b;

};

Calc.prototype.Subtraction = function (a, b) {

return a - b;

};

Calc.prototype.Multiply = function (a, b) {

return a \* b;

};

Calc.prototype.Division = function (a, b) {

return a / b;

};

return Calc;

}());

// Создание экземпляров классов

var rect = new Rectangle(6, 6);

var circ = new Circle(9);

var student = new Student('Иванов Иван', 12, 5.5);

var calc = new Calc();

// Получение элемента для вывода

var output = document.getElementById('outputSpan');

// Вывод информации о прямоугольнике

output.innerHTML += "\u041F\u0440\u044F\u043C\u043E\u0443\u0433\u043E\u043B\u044C\u043D\u0438\u043A: <br> \u041F\u043B\u043E\u0449\u0430\u0434\u044C: ".concat(rect.calcArea(), " <br> \u041F\u0435\u0440\u0438\u043C\u0435\u0442\u0440: ").concat(rect.calcPerimeter(), " <br><br>");

// Вывод информации о круге

output.innerHTML += "\u041A\u0440\u0443\u0433: <br> \u041F\u043B\u043E\u0449\u0430\u0434\u044C: ".concat(circ.calcArea(), " <br> \u041F\u0435\u0440\u0438\u043C\u0435\u0442\u0440: ").concat(circ.calcPerimeter(), " <br><br>");

// Вывод информации о студенте

output.innerHTML += "\u0421\u0442\u0443\u0434\u0435\u043D\u0442: <br> Name: ".concat(student.getName(), " <br> \u0412\u043E\u0437\u0440\u0430\u0441\u0442: ").concat(student.getAge(), " <br> \u0421\u0440\u0435\u0434\u043D\u0438\u0439 \u0431\u0430\u043B\u043B: ").concat(student.getGrades(), " <br><br>");

// Изменение информации о студенте

student.setName('Петров Петр');

student.setAge(23);

student.setGrades(5.7);

// Вывод измененной информации о студенте

output.innerHTML += "\u0418\u0437\u043C\u0435\u043D\u0435\u043D\u043D\u0430\u044F \u0438\u043D\u0444\u043E\u0440\u043C\u0430\u0446\u0438\u044F \u043E \u0441\u0442\u0443\u0434\u0435\u043D\u0442\u0435: <br> Name: ".concat(student.getName(), " <br> \u0412\u043E\u0437\u0440\u0430\u0441\u0442: ").concat(student.getAge(), " <br> \u0421\u0440\u0435\u0434\u043D\u0438\u0439 \u0431\u0430\u043B\u043B: ").concat(student.getGrades(), " <br><br>");

// Вывод результатов работы калькулятора

output.innerHTML += "\u041A\u0430\u043B\u044C\u043A\u0443\u043B\u044F\u0442\u043E\u0440: <br> \u0421\u043B\u043E\u0436\u0435\u043D\u0438\u0435 (1+0): ".concat(calc.Addition(1, 0), " <br> \u0412\u044B\u0447\u0438\u0442\u0430\u043D\u0438\u0435(1-0): ").concat(calc.Subtraction(1, 0), " <br> \u0423\u043C\u043D\u043E\u0436\u0435\u043D\u0438\u0435(2\*2): ").concat(calc.Multiply(2, 2), " <br> \u0414\u0435\u043B\u0435\u043D\u0438\u0435(4/2): ").concat(calc.Division(4, 2), " <br><br>");

1. **Вывод:**

Были сформированы практические навыки работы с прототипами и классами в JavaScript.