ГУАП

КАФЕДРА № 41

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | Н. И. Мирошниченко |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №3 |
| Практическое задание №3 по курсу: ОСНОВЫ FRONTEND-РАЗРАБОТКИ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ гр. № | 4217 |  |  |  | В. А. Милованов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2025

**Цель работы**

Формирование практических навыков работы с прототипами и классами в JavaScript.

**Ход работы**

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

С помощью ключевого слова class был создан класс Figure с двумя необходимыми для переопределения методами. Далее реализовывались классы Rectangle и Circle с использованием extends для наследования свойств от основного класса Figure. Также использовалось ключевое слово super() для вызова конструктора родительского класса. Результат вызова методов созданных классов был отображен на HTML странице методом alert() (см. рис. 1-3).

****

Рисунок 1 – Реализация класса Figure



Рисунок 2 – Реализация классов Circle и Rectangle с проверкой методов

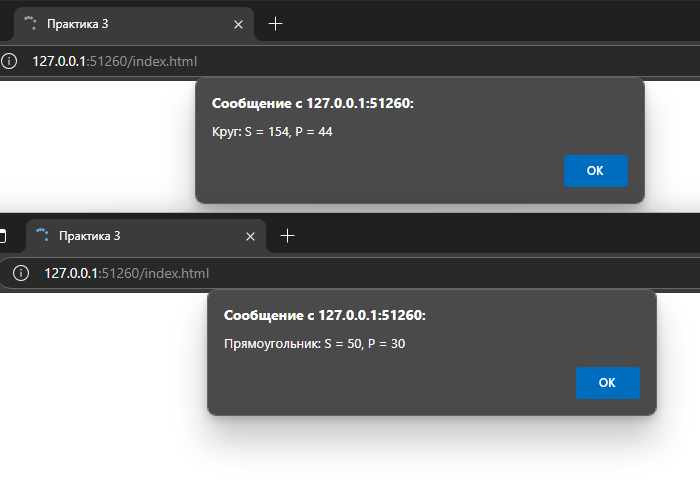


Рисунок 3 – Результат вывода

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

Аналогичным образом производилась реализация класса Student с добавлением префикса приватных полей # для защиты свойств. Были созданы особые методы - геттеры (getName(), getAverageGrade(), getAge()) и сеттеры (setName(), setAverageGrade(), setAge()) для получения и изменения заданных свойств (см. рис. 4-6).



Рисунок 4 – Реализация класса Student

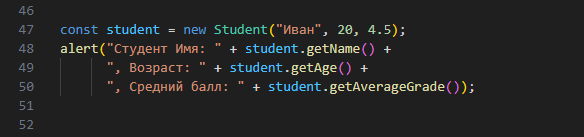


Рисунок 5 – Проверка методов класса Student

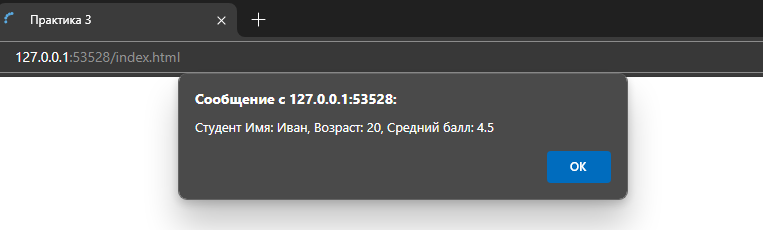


Рисунок 6 – Вывод результатов проверки Student

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

Был реализован и проверен класс Calculator с 4 методами, соответствующими 4 основным арифметическим операциям (см. рис. 7-9).

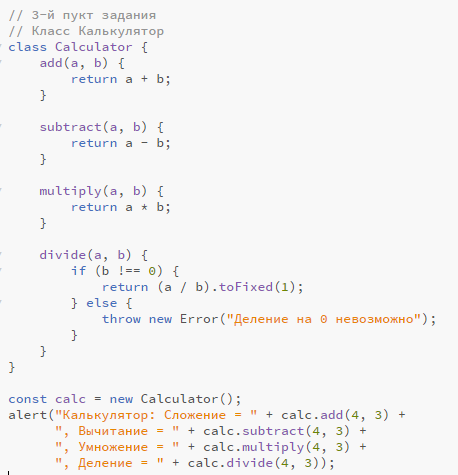


Рисунок 7 – Реализация класса Calculator

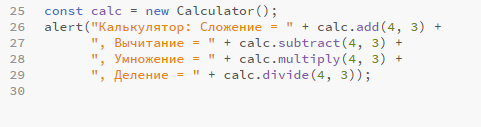


Рисунок 8 – Проверка методов класса Calculator

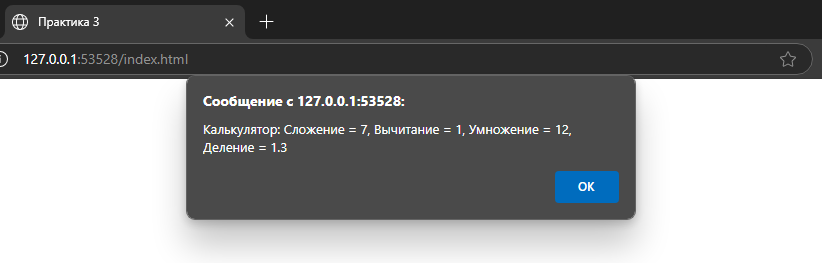


Рисунок 9 – Вывод результатов проверки Calculator

**Вывод**

В ходе выполнения практической работы были реализованы три различных класса, демонстрирующих основные принципы объектно-ориентированного программирования в JavaScript: наследование, инкапсуляцию и полиморфизм.

Класс Figure был создан как базовый, а его методы getArea() и getPerimeter() были переопределены в производных классах Rectangle и Circle, что продемонстрировало механизм наследования и полиморфизма. Это позволило использовать единые методы для различных фигур, обеспечивая удобную и гибкую архитектуру.

Класс Student был разработан с использованием инкапсуляции, что обеспечило защиту данных при помощи приватных свойств (#name, #age, #averageGrade). Доступ к этим данным осуществлялся через геттеры и сеттеры, предотвращая некорректные изменения извне.

Класс Calculator реализовал базовые математические операции (сложение, вычитание, умножение и деление), демонстрируя принципы модульности и повторного использования кода.

Таким образом, в ходе работы была создана объектно-ориентированная архитектура, обеспечивающая надежность, удобство использования и гибкость кода. Реализованные классы могут быть использованы в реальных проектах, а примененные принципы ООП делают код структурированным, читаемым и легким для расширения.