

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА  
(САМАРСКИЙ УНИВЕРСИТЕТ)»

Институт Информатики и кибернетики   
Кафедра Программных систем

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**  
  
к лабораторной работе №3 по дисциплине «Прикладные информационные системы»

Обучающийся группы 6132-020402D \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дьяконов А.В.

Обучающийся группы 6132-020402D \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лазарев М.Ю.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лобанков А.А.

Самара 2024

СОДЕРЖАНИЕ

[1 Постановка задачи 3](#_Toc180327845)

[2 Результаты работы 4](#_Toc180327846)

[2.1 Результаты задания 1 4](#_Toc180327847)

[2.2 Результаты задания 2 4](#_Toc180327848)

[2.3 Результаты задания 3 4](#_Toc180327849)

[2.4 Результаты задания 4 5](#_Toc180327850)

[ПРИЛОЖЕНИЕ А Листинг программы 7](#_Toc180327851)

1. Постановка задачи

В процессе выполнения заданий ознакомиться с циклом событий, замыканиями и декораторами в TS.

1. Результаты работы
   1. Результаты задания 1

Реализуйте функцию удаления нечетных чисел из одномерного массива натуральных чисел. После удаления одного из таких чисел, следует подождать 1 секунду до удаления следующего. Операции удаления следует показывать в консоли.

1. Создаем функцию для задержки delay с помощью Promise, который разрешается через указанное количество миллисекунд.
2. Создаем асинхронную async функцию removeOddNumbers, которая удаляет все нечетные числа из массива numbers и использует delay для задержки между удалениями чисел на 1 секунду.
3. Выводится результат согласно примеру (Рис. 1)(Рис. 2).
   1. Результаты задания 2

Реализуйте на ваш выбор функцию, которая возвращает функцию, которая в свою в свою очередь пишет результат в консоль некоторую строку. Вызвать данные функции, показав принцип работы замыканий.

1. Создаем функцию createLock, которая создает замыкание, возвращая функцию, которая выводит сообщение.
2. Возвращаемая функция "замыкает" переменную message.
3. Вызываем функции и выводим результат согласно примеру (Рис. 1)(Рис. 2).
   1. Результаты задания 3

Создайте декоратор, выполняющий блокировку изменения прототипа класса автомобиль. Необходимо проверить, осталась ли возможность добавления сторонних полей в объект после введения декоратора (проверка работоспособности).

1. Создаем декоратор класса FreezeClass, который замораживает как сам конструктор, так и его прототип.
2. Используем @FreezeClass с классом Автомобиль MyCarClass, который предотвращает добавление новых свойств как в экземпляры класса, так и в его прототип после инициализации.
3. Проверяем работоспособность, добавляя новые поля в объект класса и прототип, и получаем результаты (Рис. 1)(Рис. 2).
   1. Результаты задания 4

Реализуйте декоратор метода, выполняющий преобразование возвращаемой строки с некими сведениями об объекте класса Автомобиль, заменяя все буквы на ЗАГЛАВНЫЕ.

1. Создаем декоратор метода upperCase1, преобразующий результат в заглавные буквы.
2. Переопределяем дескриптор для использование toUpperCase.
3. Применяем декоратор @upperCase1 к методу getCarInfo() для класса My2CarClass.
4. Получаем вывод примера в верхнем регистре (Рис. 1)(Рис. 2).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 1 – Результаты выполнения заданий в Google Chrome

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок – Результат выполнения заданий в VS Code

ПРИЛОЖЕНИЕ А  
Листинг программы

// Задание 1

let delay = (ms: number) => new Promise(resolve => setTimeout(resolve, ms));

async function removeOddNumbers(numbers: number[]): Promise<number[]> {

let index = 0;

while (index < numbers.length) {

if (numbers[index] % 2 !== 0) {

console.log(`Удаление нечетного числа: ${numbers[index]}`);

numbers.splice(index, 1);

await delay(1000);

} else {

index++;

}

}

return numbers;

}

// Пример использования

let numberss = [1, 2, 3, 4, 5, 6, 7, 8, 9];

removeOddNumbers(numberss).then(result => {

console.log("Числа:", result);

});

// Задание 2

function createLock(message: string): () => void {

return function() {

console.log(`Замыкание: ${message}`);

};

}

let lock1 = createLock("Раз");

let lock2 = createLock("Два");

// Вызов функций

lock1();

lock2();

// Задание 3

function FreezeClass(constructor: Function) {

console.log("Freeze decorator");

Object.freeze(constructor);

Object.freeze(constructor.prototype);

}

@FreezeClass

class MyCarClass {

public brand: string;

public model: string;

public year: number;

public vin: string;

public regNumber: string;

constructor(

brand: string,

model: string,

year: number,

vin: string,

regNumber: string,

) {

this.brand = brand;

this.model = model;

this.year = year;

this.vin = vin;

this.regNumber = regNumber;

}

printInfo(): void {

console.log(`Автомобиль: ${this.brand} ${this.model},

Год: ${this.year}, VIN: ${this.vin},

Регистрационный номер: ${this.regNumber}`);

}

}

//Проверка работоспособности

let MyCarClass1 = new MyCarClass("Lada", "Granta", 2013, "1234", "567890");

MyCarClass1.printInfo();

(MyCarClass1 as any).newField = "Новое поле";

console.log(MyCarClass1);

(MyCarClass.prototype as any).newPrototypeField = "Новое поле в прототипе";

console.log(MyCarClass.prototype);

console.log((MyCarClass as any).newField); //

console.log((MyCarClass.prototype as any).newPrototypeField);

// Задание 4

function upperCase1(target:any, key:string, descriptor:PropertyDescriptor){

// Переопределяем дескриптор свойства

Object.defineProperty(target, key, {

...descriptor,

value: function() {

return descriptor.value.call(this).toUpperCase();

}

}

);

return target;

}

class My2CarClass extends MyCarClass {

constructor(

brand: string,

model: string,

year: number,

vin: string,

regNumber: string

) {

super(brand, model, year, vin, regNumber);

}

@upperCase1

getCarInfo(): string {

return `Автомобиль: ${this.brand} ${this.model}, Год: ${this.year},

VIN: ${this.vin}, Регистрационный номер: ${this.regNumber}`;

}

}

// Пример использования

let MyCarClass2 = new My2CarClass("Lada", "Granta", 2013, "1234", "567890");

console.log(MyCarClass2.getCarInfo());