

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 №1 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](#_Toc178189259)

[2 Результаты работы 4](#_Toc178189260)

[2.1 Результаты задания 1 4](#_Toc178189261)

[2.2 Результаты задания 2 4](#_Toc178189262)

[2.3 Результаты задания 3 5](#_Toc178189263)

[2.4 Результаты задания 4 5](#_Toc178189264)

[2.5 Результаты задания 5 6](#_Toc178189265)

[2.6 Результаты задания 6 7](#_Toc178189266)

[ПРИЛОЖЕНИЕ А Листинг программы 8](#_Toc178189267)

1. Постановка задачи

Ознакомиться с использованием структуры данных, массивов, кортежей, дженериков, перечислений и кастомных типов в языке TypeScript.

1. Результаты работы
   1. Результаты задания 1

Реализовать метод, возвращающий максимальное значение среди отрицательных чисел массива целых чисел.

1. Создаем функцию getMaxNegativeValue, которая имеет параметр arr и принимает массив чисел arr:number [] в качестве аргумента.
2. Фильтруем массив методом filter, чтобы у нас остались только отрицательные числа. Для этого создается новый массив negativeNumbers с условием num<0.
3. Проводим проверку на отсутствие отрицательных чисел с использованием if. Если длина массива negativeNumbers равна нулю, то функция возвращает null.
4. При наличии отрицательных чисел метод Math.max возвращает максимальное значение из заданного массива.
5. Получаем результат (Рис. 1) согласно выбранному примеру.



Рисунок 1 - Результат задания 1

* 1. Результаты задания 2

Реализовать метод, возвращающий матрицу целых чисел, составленный из матрицы булевых значений (значение true переходит в формат числа 1, а false в формат числа 0).

1. Создаем функцию booleanMatrixToIntMatrix. Она принимает двумерный массив (booleanMatrix), в котором каждый элемент является массивом булевых значений (true или false).
2. Функция возвращает новый двумерный массив чисел. Метод map используется дважды:

Внешний map проходит по каждой строке (row) булевой матрицы.

Внутренний map преобразует каждое булево значение value в число.

Если значение true, оно становится 1; если false, оно становится 0. Это делается с помощью оператора: value ? 1 : 0.

1. Получаем результат (Рис. 2) согласно выбранному примеру.



Рисунок 2 – Результат задания 2

* 1. Результаты задания 3

Создайте кортеж, который может содержать только 2 строковых и 1 числовое значения (порядок следования: число,строка0, строка1). Реализуйте метод, возвращающий конкатенацию строк в формате: строка1: строка0

1. Создаем пользовательский тип кортежа type MyTuple, указываем содержание: число number, строка string, строка string.
2. Применяем метод для конкатенации. Создаем функцию concatenateStrings, которая принимает кортеж типа MyTuple. Используем деструктуризацию массива для извлечения элементов кортежа по порядку: число num, строка 0 str0, строка 1 str1.

Следом используем форматирование строки с использованием шаблонов ${str1}: ${str0}.

1. Получаем результат (Рис. 3) согласно выбранному примеру.



Рисунок 3 – Результат задания 3

* 1. Результаты задания 4

Создайте тип перечисление для типов стандартных аминокислот (Глицин, Глутамин и т.д). Выведите какой-либо тип аминокислоты в консоль.

1. Используем перечисление enum, при этом сгруппированные константы будут по отдельности ассоциироваться с конкретным строковым значением.
2. Обратимся к элементу с помощью точечной нотации AminoAcid.Glicine.
3. Получаем результат (Рис. 4) согласно выбранному примеру.



Рисунок 4 – Результат задания 4

* 1. Результаты задания 5

Используя следующий код, реализуйте метод, который будет выводить информацию в консоль о создаваемом объекте типа

Cat или Dog, применяя обобщенный тип, ограниченный типом Pet.

1. Применяем обобщенную функцию printPetInfo, которая имеет ограничение классом Pet <T extends Pet>. Любой объект, переданный в эту функцию, должен быть либо объектом типа Pet, либо его потомком (Cat, Dog).
2. Выводится имя, возраст и результат выполнения метода speak() для любого объекта класса Pet или его потомков. Если объект является собакой (instanceof Dog), выводится дополнительная информация — поле label.
3. Получаем результат (Рис. 5).

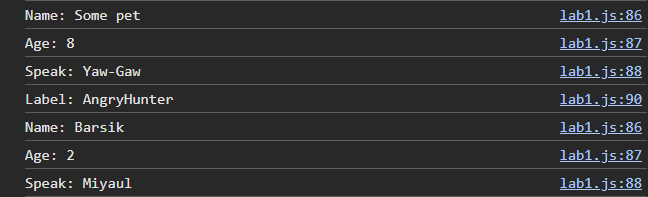


Рисунок 5 – Результат задания 5

* 1. Результаты задания 6

Создайте тип с применением перечисления из задания (для использования его в качестве типа поля, для некоторых случаев возможно его использование при реализации массива). Добавьте собственные поля стандартных типов, корректно характеризующие ту или иную предметную область, совпадающую с вашим типом перечисления. Создайте объект на основе вашего типа и выведите его в консоль в формате JSON.

1. Берём перечисление аминокислот из задания и создаём тип для описания AminoАcidInfo, в котором определяем поля название name, трехбуквенный код code и молекулярную массу molecularWeight.
2. Создаём объект glycineInfo, который соответствует типу AminoAcidInfo, где указываем информацию по глицину в соответствующих полях.
3. В результате выводится JSON строка методом JSON.stringify (Рис. 6).



Рисунок 6 – Результат задания 6.

ПРИЛОЖЕНИЕ А  
Листинг программы

// Задание 1

function getMaxNegativeValue(arr: number[]): number | null {

    const negativeNumbers = arr.filter(num => num < 0);

    if (negativeNumbers.length === 0) {

        return null;

    }

    return Math.max(...negativeNumbers);

}

// Пример

let numbers = [-10, -20, -3, -50, -7];

let maxNegative = getMaxNegativeValue(numbers);

console.log(maxNegative);

// Задание 2

function booleanMatrixToIntMatrix(booleanMatrix: boolean[][]): number[][] {

    return booleanMatrix.map(row => row.map(value => value ? 1 : 0));

}

// Пример

let booleanMatrix = [

    [true, false, true],

    [false, true, false],

    [true, true, false]

];

let intMatrix = booleanMatrixToIntMatrix(booleanMatrix);

console.log(intMatrix);

// Задание 3

type MyTuple = [number, string, string];

const myTuple: MyTuple = [42, "Hello", "World"];

function concatenateStrings(tuple: MyTuple): string {

    const [num, str0, str1] = tuple;

    return `${str1}: ${str0}`;

}

// Пример

let result = concatenateStrings(myTuple);

console.log(result); // Вывод: World: Hello

// Задание 4

enum AminoAcid {

    Glycine = "Глицин",

    Glutamine = "Глутамин",

    Alanine = "Аланин",

    Isoleucine = "Изолецитин",

    Phenylalanine = "Фенилаланин",

    Tyrosine = "Тирозин",

    Tryptophan = "Триптофан",

    Asparagine = "Аспаргин",

    Arginine = "Аргинин",

}

// Пример

console.log(AminoAcid.Glycine);

// Задание 5

class Pet {

name = 'Some pet';

age = -1;

speak() {

return "No speak. I am fish!";

}

}

class Dog extends Pet {

label = 'AngryHunter';

age = 8;

speak() {

return "Yaw-Gaw";

}

}

class Cat extends Pet {

name = 'Barsik';

age = 2;

speak() {

return "Miyaul";

}

}

function printPetInfo<T extends Pet>(pet: T): void {

    console.log(`Name: ${pet.name}`);

    console.log(`Age: ${pet.age}`);

    console.log(`Speak: ${pet.speak()}`);

    if (pet instanceof Dog) {console.log(`Label: ${pet.label}`)}

}

// Пример

let myCat = new Cat();

let myDog = new Dog();

printPetInfo(myDog); // Информация о собаке

printPetInfo(myCat); // Информация о коте

// Задание 6

type AminoAcidInfo = {

    name: AminoAcid;

    code: string;

    molecularWeight: number;

};

let glycineInfo: AminoAcidInfo = {

    name: AminoAcid.Glycine,

    code: "GLY",

    molecularWeight: 75.07

};

console.log(JSON.stringify(glycineInfo, null, 2));