Задача 1

export type User = any;

export const users: any[] = [

{

name: 'Max Mustermann',

age: 25,

occupation: 'Chimney sweep'

},

{

name: 'Kate Müller',

age: 23,

occupation: 'Astronaut'

}

];

export function logPerson(user: any) {

console.log(` - ${user.name}, ${user.age}`);

}

console.log('Users:');

users.forEach(logPerson);

Задача 2

type User = {

id: string;

name: string;

}

type Course = {

id: number;

title: string;

}

type WithRate = {

rate: 1 | 2 | 3 | 4 | 5;

}

type WithStudentRole = {

role: "student"

}

type WithTeacherRole = {

role: "teacher"

}

type WithLevel = {

level: "junior" | "middle" | "senior"

}

/\* --- \*/

type StudentCourse = Course & WithStudentRole & WithRate & WithLevel

type Student = any & { courses: { [id: number]: StudentCourse } }

type Teacher = any

type Director = any;

/\*-- Проверка --\*/

const s1: Student = {

id: "s1",

name: "s1",

courses: {

[1]: {

id: 1,

title: "First",

rate: 5,

role: "student",

level: "middle"

}

},

}

const t1: Teacher = {

id: "t1",

name: "t1",

courses: {

[5]: {

level: "junior",

id: 5,

title: "Fifth",

role: "teacher"

},

[1]: {

...s1.courses[1],

role: "teacher"

}

}

}

const d1: Director = {

id: "d1",

name: "d1",

students: {

["s1"]: s1,

["s2"]: {

id: "s2",

name: "s2"

}

},

teachers: {

["t1"]: {

...t1,

rate: 3,

},

["t2"]: {

id: "t2",

name: "t2",

level: "senior",

rate: 5

}

}

}

Задание 3

function zip<T, U>(first: T[], second: U[]): Array<[T, U]> {

const minLength = Math.min(first.length, second.length);

const result: Array<[T, U]> = [];

for (let i = 0; i < minLength; i++) {

result.push([first[i], second[i]])

}

return result;

}

const q1: Array<[number, string]> = zip([1, 2, 3, 4, 5, 6], ["1", "2", "3"]);

const q2: Array<[boolean, boolean]> = zip([true], [false, false]);

console.log(q1, q2);

function groupBy<T, K, V>(source: T[], keySelector: (item: T, index: number) => K, valueSelector: (item: T, index: number) => V): Map<K, V[]> {

const result = new Map<K, V[]>();

for (let i = 0; i < source.length; i++) {

const item = source[i];

const key = keySelector(item, i);

const value = valueSelector(item, i);

if (!result.has(key)) {

result.set(key, []);

}

result.get(key)?.push(value);

}

return result;

}

const q1: Map<number, number[]> = groupBy([1, 2, 3, 4], x => x % 2, x => x + 1);

const q2: Map<boolean, {x: string, i: number}[]> = groupBy(["aaa", "bbb", "cc", "q", "lalaka"], (\_, i) => i % 2 === 0, (x, i) => ({i, x}));

console.log(q1, q2);

Задача 4

type User = {

id: string;

name: string;

}

type Teacher = Partial<User>;

type Director = Partial<User>;

type Role = "student" | "teacher"

type Rate = 1 | 2 | 3 | 4 | 5

type Level = "junior" | "middle" | "senior"

type Course = {

id: number;

title: string;

role: Role;

rate: Rate;

level: Level;

}

/\* --- \*/

type Student = User & { courses: { [id: number]: Omit<Course, "role"> & { role: Exclude<Role, "teacher">} } }

type Teacher = User;

type Director = User;

/\*-- Проверка --\*/

const s1: Student = {

id: "s1",

name: "s1",

courses: {

[1]: {

id: 1,

title: "First",

rate: 5,

level: "middle"

}

},

}