

ESMAD | TSIW | POO
Teste Modelo**GRUPO I - Estruturas e funções**

1. Crie um programa que receba dois números entre 1 e 99 (inclusivé). O programa deve repetir a pergunta até que sejam digitados dois números dentro desse intervalo. Após ter os números válidos, estes devem ser passados para uma função `getNums(num1, num2)` definida como uma expressão. Esta função deve devolver a soma de todos os números desse intervalo que são pares e múltiplos de 5.

O resultado final deve ser apresentado na consola da seguinte forma usando template string: "O resultado final é de X".

EXEMPLOS:

7, 13 => O resultado final é de 10

10, 20 => O resultado final é de 30

GRUPO II - Manipulação DOM

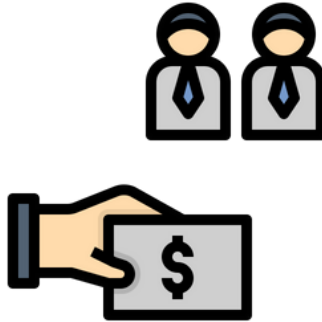
2. Crie uma tabela 3 por 3. Mapeie a tabela de forma que ao clicar em qualquer célula seja injetada uma letra do abecedário sequencial (começando pelo "A"). Ao clicar numa célula que já contém uma letra, deve ser dada uma notificação de erro.

A		
	B	
D		C

Dica: use as funções `charCodeAt` e `fromCharCode` para gerir a conversão da letra de/para uma unidade de código UTF-16

GRUPO III - Classes, objetos e arrays

3. A empresa **GoTech** está a criar uma app para gerir toda a parte financeira, mais concretamente, os seus custos com pessoal.



Complete o seguinte excerto de código que contém a definição de uma classe para modelar os salários dos funcionários de uma empresa organizada em departamentos:

```
class Company {  
    constructor() {  
        this._employees = []  
        this._departments = ['Informatics', 'Financial', 'Marketing']  
    }  
    ...  
}
```

3.1 Crie um método **hireEmployee(name, department, salary)** que permita contratar um novo empregado dado o seu nome, departamento que o contratou e salário hora. O método deve lançar erros para quem o invocou sempre que o departamento não exista ou o nome do funcionário já exista na empresa.

Depois, fora da classe, invoque o método 4 vezes com os seguintes dados:

- Manuel Rocha, Informática, 12€/hora
- Maria Bastos, Financeiro, 20€/hora
- Pedro Silva, Marketing, 10€/hora
- Silvia Rocha, Informática, 12€/hora

Não se esqueça de tratar potenciais erros.

3.2 Crie um método chamado **getEmployees()** que devolva um array com toda a informação de todos os funcionários da empresa.

3.3 Crie um novo método chamado **increaseSalary(department, value)** que aumente em **value** o salário hora dos funcionários de um dado departamento. O método deve lançar erro caso o departamento não exista.

3.4 Crie um método chamado **getWeeklyExpenseEmployees()** que deve devolver o valor gasto pela empresa para pagar os salários de todos os trabalhadores. Tenha em conta que um dia de trabalho são 8 horas e que os empregados trabalham 5 dias por semana.

GRUPO IV - Criação de app MVC

4. Crie uma pequena app chamada **myGrades** que faça a gestão das suas notas no curso de TSIW.



A UI será muito simples: para inserir uma nota crie um formulário com uma única caixa de texto e um botão de submissão. Use a estrutura **ano#UC#nota** para inserir uma nota e a visualização de todas as notas deve ficar a cargo duma tabela que automaticamente atualizará toda a informação agregadas nas seguintes colunas: ano, nome da UC, nota e ações (remoção). No final deve existir um botão **Estatísticas** que ao ser pressionado deve exibir uma modal com a média por ano e a média global.

Algumas notas:

- As UCs com notas negativas devem surgir com fundo vermelho
- Ao fechar a app todos os dados devem ser persistidos no armazenamento local
- O uso do padrão MVC é valorizado