

ESCOLA
SUPERIOR
DE MEDIA
ARTES
E DESIGN
POLITÉCNICO
DO PORTO

ALP

PROVA DE AVALIAÇÃO
ATIVIDADE LETIVA

Tecnologias e Sistemas de Informação para a Web

CURSO

2021/2022	2022/07/21	14h30	2h
-----------	------------	-------	----

ANO LETIVO

DATA

HORA

DURAÇÃO

Programação Orientada a Objetos

1º ano

UNIDADE CURRICULAR

ANO

Mário Pinto / Ricardo Queirós/Daniel Carneiro

Recurso

DOCENTE

ÉPOCA

Observações:

- O Exame é individual
- Podem consultar **apenas** as seguintes referências externas: MDN, W3Schools, Moodle e GitHub (repo da UC e repo pessoal)
- A avaliação do teste poderá implicar convocar os estudantes para uma sessão **de avaliação oral**, onde podem ter que explicar o código entregue
- Serão usadas ferramentas anti-plágio na avaliação dos testes
- **Crie uma pasta com o seu número de aluno. Guarde os exercícios nessa pasta. No final do teste submeta a pasta (compactada) no moodle, em objeto próprio de submissão**

I (25%)

Crie um programa que receba números do utilizador, sucessivamente, até que este introduza -1. Depois o programa deve devolver o número menos frequente. Em caso de empate deve devolver o primeiro que encontrar.

Para determinar o número menos frequente, deve invocar uma função de expressão **getLessFrequent(nums)** que irá devolver o número menos vezes digitado.

O programa tem de permitir a introdução de pelo menos um número antes de terminar a introdução de valores (ao introduzir -1).

O resultado deve ser apresentado numa caixa de alerta da seguinte forma, usando template string: "O número menos frequente é o N". Sendo N o número menos frequente.

EXEMPLOS:

2 3 2 3 4 => O número menos frequente é o 4

3 2 0 5 => O número menos frequente é o 3

7 => O número menos frequente é o 7

II (25%)

Crie uma tabela com 3 linhas e 3 colunas. Mapeie a tabela de forma que ao clicar em qualquer célula, seja injetada uma letra do abecedário (a letra deve ser sorteada aleatoriamente, a partir do alfabeto, mas nunca repetida). Ao clicar numa célula que já contém uma letra deve ser dada uma notificação de erro.

A		
	B	
D		C

III (50%)

A app **Gameland** permite fazer a gestão dos seus jogos preferidos. A app tem uma classe principal que faz a gestão central destes dados.



Crie um ficheiro **gotechModel.js** com uma classe que modele um funcionário da empresa. Um funcionário é caracterizado por um nome, departamento (note que um funcionário pode trabalhar em mais do que um departamento), nº horas de trabalho, valor/hora, e um array com os nomes das empresas onde prestou serviços.

- a. Crie a classe **employee**
 - b. Crie um construtor passando todos os parâmetros necessários. Tome nota que se não for passado o parâmetro valor/hora, deve ser considerado por defeito o valor de 30€ por hora.
 - c. Crie as propriedades privadas com os respetivos métodos **get** e **set**. Tome nota que o número de horas de trabalho não deve ultrapassar nunca as 140 h (trata-se do valor mensal, máximo de 140h por mês). Caso contrário deve ser gerado um erro **“Nº máximo de horas de trabalho já atingido”**.
 - d. Crie um método **addHours** que deve adicionar horas de trabalho (nome do funcionário, e nº de horas recebidos por parâmetro), caso não tenha atingido o número máximo de horas.
 - e. Crie um método **editCompanies** que deve acrescentar uma empresa à lista de empresas onde um funcionário prestou serviços (nome do funcionário e nome da empresa recebidos por parâmetro).
2. Crie um ficheiro **gotechView.js** que deve importar a classe anterior e deve ter as seguintes funcionalidades:
- a. Crie 3 funcionários (3 instâncias da classe employee):
 - i. "Carlos Andrade",
["Informatica", "Electronica"],
12,
35,
["MKTalent", "F3M "]
 - ii. "Maria Pereira ",
["Informatica", "Redes"],
20,
25,
["SKRey"]
 - iii. "Carla Fonseca",
["Marketing"],
25,
30,
["SKRey", "MKTalent "]

- b. Adicione os funcionários a um array **employees**
- c. Crie as seguintes funções que devem ser invocadas por botões (F1 a F5) na página HTML. Os parâmetros devem ser passados pelo utilizador através da função *prompt*. Os resultados devem surgir em caixas de alerta:
- Devolver o número total de horas de trabalho de todos os funcionários
 - Devolver o valor total a pagar a um determinado funcionário (nome indicado através da função *prompt*)
 - Devolver os nomes das empresas para as quais um determinado funcionário prestou serviço (nome do funcionário indicado através da função *prompt*)
 - Devolver a nome do funcionário com mais horas de trabalho
 - Adicionar empresa e nº de horas de trabalho a um funcionário. A função deve receber o nome do funcionário, nº de horas e nome empresa (inseridos por *prompt*). A função deve somar o nº de horas de trabalho ao funcionário (desde que não ultrapasse as 140h). Caso ultrapasse, deve ser lançada uma mensagem de erro. A função deve ainda verificar se a empresa já existe na instância desse funcionário. Caso não exista, deve ser acrescentada ao array.
3. No ficheiro **index.html** deve referenciar o ficheiro anterior como um módulo e deve incluir buttons para as 5 funções acima referidas.
- Os dados da tabela devem ser renderizados programaticamente (no ficheiro JS), e devem estar ordenados por nome de funcionário.

FUNÇÕES				
F1	F2	F3	F4	F5
Funcionários				
NOME	DEPARTAMENTOS	Nº HORAS	V.H	TOTAL
Carla Fonseca	[Marketing]	25	30	750
Carlos Andrade	[Informatica, Eletronica]	12	35	420
Maria Pereira	[Informatica, Redes]	20	25	500
Total: 1670				

