

DAM – M03 – Programació

UF2 – Programació modular

Utilitza el disseny modular per a donar una solució als següents problemes. Alhora d'avaluar les solucions es tindrà en consideració l'ús correcte del disseny modular en les solucio

Problema 1. (20%) Escriu un mètode que rebí un string i un caràcter i calculi el número d'aparicions del caràcter dins de l'string. Implementa el mètode de dues maneres diferents: com una funció anomenada numCharFunction (que retorni el número d'ocurrències del caràcter) i com una acció anomenada numCharAction (que no retorni res i escriui el número d'ocurrències per pantalla dins la funció).

Problema 2. (40%) En una empresa de transport en han demanat que desenvolupem un programa molt senzill que sumi els kms que fa la seva flota de camions.

Ens han donat les següents especificacions que cal incloure en un petit programa capaç de comptar kms.

“El programa s’inicialitza dins el main i es demana a l’usuari el nombre de camions que té la flota. A continuació es demana que implementeu les diferents opcions dins un menú:

- introduirDades
- kmsCamio
- llistatKmCamioMes
- kmsTotals

I aquest n’és el Contracte amb l’empresa (són els acords que s’estableixen, les funcions i accions han de tenir aquestes signatures):

- *public static void introduirDadesDia(int[] kmsCamioDia):* Aquesta acció ha d’introduir per cada dia els kms que fa cada camió durant un mes. (Aquesta informació es guardarà en una matriu de dades kmsCamionsDia on les files seran els camions i les columnes els dies d’un mes.
- *public static int kmsCamio(int[] kmsCamioDia, int camio):* Aquesta funció retorna tots els kms que fa un camió durant un mes.
- *public static int[] llistatKmCamioMes(int[] kmsCamioDia, int camio):* Aquesta funció retorna el llistat de km que efectua un camió durant un mes.
- *public static int kmsTotals(int[] kmsCamioDia):* Aquesta funció retorna tots els kms que fan tots els camions durant un mes.”

Dins de cada opció de menú imprimeix el resultat que retorna la funció del contracte. Podeu crear els mètodes auxiliars que necessiteu.

Problema 3. (40%)

Repartició de vacunes:

El govern central ens ha demanat un programa per controlar la distribució de vacunes. Aquest programa ha de contemplar la repartició de **17 containers**, un per cada comunitat autònoma, cada container tindrà un tamany de 10.000 vacunes.

Es demana un petit programa que ajudi al la seva distribució amb diferents opcions:

- Els containers inicialment es troben a la seu central i estan buits.
- Permet **introduir** vacunes en qualsevol container indicant el nombre de vacunes introduïdes.
- Permet **treure** vacunes d'un container.
- Permet **moure** vacunes entre containers.
- Veure l'estat de qualsevol container (es mostra quants de les 10.000 posicions del container estan plenes).

Pista: **Controlar els límits** alhora de treure o posar vacunes dins un container.

Problema Bonus (+1 punt)

Recursivitat (2p): Realitza un programa que donat un nombre N, mostri els nombres des de 1 fins a N que siguin divisibles per 3.

Per exemple: Si en el programa introdueixo el 100 m'ha de mostrar i en aquest ordre: 3,6,9,12, ... , 90,93,96,99.

La capçalera del programa és: `public static void escriuMultiplesTres(int N)`

Evidentment: Només es tindran en compte les respostes realitzades sota una metodologia top-down amb l'especificació correcta de les funcions/accions emprades.

NOTA: L'ideal en totes tres activitats és que en el main només hi hagi crides a funcions que heu desenvolupat.