

CFGS Desenvolupament d'Aplicacions Multiplataforma
CFGS Desenvolupament d'Aplicacions Web
M3 – Programació bàsica (Bloc 1)

PROVA D'AVALUACIÓ FINAL_TELEMÀTICA 1
Curs 2020-21 – Semestre 1

Fitxer de la prova

• **Durada total de la prova: 150'**

• **Codi** (adjunt al final de la pàgina. **És l'aplicació que teníeu com a model de solució del projecte**)

• **Aplicació per l'entrada de dades dels alumnes**

Enunciat (model A)

Modifiqueu l'aplicació per a l'entrada de les dades dels alumnes. Heu d'afegir les següents funcionalitats:

1- Completar l'aplicació perquè també demani l'edat. Les edats entrades cal que es guardin sobre un array. Els valors vàlids per a l'edat són entre 16 i 120. Ambdós inclosos. En cas de que el valor introduït sigui incorrecte, cal donar **dos** intents extres per aquest camp. Si els esgota, cal anar a introduir dades per un nou usuari, descartant les dades corresponents.

2- A continuació, cal calcular l'edat mitjana per cada cicle. Un cop entrades totes les dades, caldrà mostrar la informació de l'edat mitjana i el número d'estudiants matriculats per a cada cicle. **Feu això només pels cicles de la família de informàtica.**

Un exemple de sortida per pantalla seria el següent:

DNI	nota	edat	cicle
78788989	7	16	Adm. de sistemes informàtics
44556677	9	22	Administració i finances
55245677	8	35	Gestió administrativa
11223344	4	19	Administració i finances

Enunciat (model B)

Modifiqueu l'aplicació per a l'entrada de les dades dels alumnes. Heu d'afegir les següents funcionalitats:

1- Completar l'aplicació perquè també demani l'any d'inici dels estudis. Els valors vàlids per l'any d'inici son entre 1992 i 2020. Ambdós inclosos. En cas de que el valor introduït sigui incorrecte, cal donar dos intents extres per aquest camp. Si els esgota, cal anar a introduir dades per al nou usuari descartant les dades corresponents.

2- Cal que feu les modificacions necessàries perquè totes les dades quedin recollides en un sol array bidimensional. Podeu, si cal, modificar el tipus d'alguna dada.

3- A continuació, cal calcular l'edat mitjana per cada cicle. Un cop entrades totes les dades, caldrà mostrar la informació de l'antiguitat mitjana i el número d'estudiants matriculats per a cada cicle. Feu això només pels cicles de la família de informàtica.

L'antiguitat es calcula restant l'any actual de l'any inicial. Per exemple, si es va iniciar estudis a l'any 2006 l'antiguitat seria $2020 - 2006 = 14$.

4- En acabar, cal donar l'opció d'ordenar les dades. Si l'usuari aixó ho vol, cal mostrar un llistat ordenat per any d'inici dels estudis.

Un exemple de sortida per pantalla seria el següent:

DNI	nota	any inici	cicle
78788989	7	2012	Adm. de sistemes informàtics
44556677	9	2001	Administració i finances
52245677	8	1992	Gestió administrativa
11223344	4	2018	Administració i finances

Lliurament

1- La carpeta del projecte amb el codi de la vostra solució dels problemes plantejats.

Comprimiu la carpeta en un sol arxiu (.zip) i aquest sigui l'arxiu que ens lliureu.

2- Un vídeo de captura de pantalla on aparegueu vosaltres i el codi que heu fet, i expliqueu quina ha estat la vostra solució.

Adjunteu l'enllaç al vídeo.

Us recomanem que feu servir l'eina loom.com per a fer el vídeo, doncs us permet fer l'enregistrament des del mateix navegador (google chrome), amb molta facilitat. Si voleu fer servir qualsevol altre eina no hi ha cap problema. Només que tingueu present que **volem veure el vostre codi i a vosaltres mentre ens ho expliqueu.**

```

*
* Aplicació per l'entrada de dades d'alumnes
*/

import java.util.Scanner;

public class InscIOC_arrays_PAF1920S2 {
    public static final int MINIM_DNI = 1;
    public static final int MAXIM_DNI = 99999999;
    public static final int MINIM_FAMILIA = 1;
    public static final int MAXIM_FAMILIA = 4;
    public static final int FAMILIA_INFORMATICA = 1;
    public static final int FAMILIA_ADMINISTRACIO = 2;
    public static final int MINIM_CICLE = 100;
    public static final int MAXIM_CICLE = 400;
    public static final int SMX = 111;
    public static final int DAM = 121;
    public static final int ASX = 122;
    public static final int DAW = 123;
    public static final String SMX_NOM = "Sistemes microinformàtics i xarxes";
    public static final String DAM_NOM = "Desenv. d'aplicacions
multiplataforma";
    public static final String ASX_NOM = "Administració de sistemes
informàtics";
    public static final String DAW_NOM = "Desenvolupament d'aplicacions web";
    public static final int GAD = 211;
    public static final int AFI = 221;
    public static final int ADI = 222;
    public static final String GAD_NOM = "Gestió administrativa";
    public static final String AFI_NOM = "Administració i finances";
    public static final String ADI_NOM = "Assistència a la direcció";
    public static final int MINIM_NOTA = 0;
    public static final int MAXIM_NOTA = 10;
    public static final int MAX_INTENTS = 3;

    public static final int MAX_ALUMNES = 4;

    public static void main(String[] args) {

        //Definim tipus de dades
        // En el cas dels arrays: declarem i reservem espai de memòria per
        // emmagatzemar dades
        int []dni = new int[MAX_ALUMNES];
        int []familia = new int[MAX_ALUMNES];
        int []cicle = new int[MAX_ALUMNES];
        int intents=0;
        int i=0;
        int seguir=0;
        float fcanvi;
        int icanvi;
        float []nota= new float[MAX_ALUMNES];
        String nomCicle="";
        Scanner scan = new Scanner(System.in);
        boolean dadaCorrecta=true, introduirMes;
        do{
            do{
                introduirMes=false; //per defecte, no volem introduir més
                //Llegim i mirem si és correcta
                System.out.print("Entra el DNI (8 xifres numèriques): ");

                dadaCorrecta = scan.hasNextInt();//per analitzar tipus de dada
                if ( dadaCorrecta ){ // si és un número sencer
                    //Llegim dades
                    dni[i] = scan.nextInt();

```

```

        //Avaluem si és vàlida
        if((dni[i] < MINIM_DNI)|| (dni[i] > MAXIM_DNI)){
            dadaCorrecta = false;
        }
    }
    intents++;
} while ((!dadaCorrecta)&&(intents<MAX_INTENTS));

if(dadaCorrecta){
    intents=0; //Per a cada dada inicialitzem el comptador d'intents
a 0
    //Entrem Cicle
    System.out.print("\n\tinformàtica i comunicacions (1)\n" +
        "\tadministració i gestió (2)\n" +
        "\telectricitat i electrònica (3)\n" +
        "\tsanitat (4)\n" +
n");
        "\tserveis socioculturals i a la comunitat (5)\n");
    do{
        System.out.print("\nEntra la familia professional:");
        dadaCorrecta = scan.hasNextInt();
        if ( dadaCorrecta ){
            familia[i] = scan.nextInt();
            if((familia[i] < MINIM_FAMILIA)|| (familia[i] >
FAMILIA_ADMINISTRACIO)){
                dadaCorrecta = false;
            }else{//Si aribem aquí es que la dada familia és
correcta
                intents=0; //Inicialitzem el comptador d'intents a 0
                switch (familia[i]){
                    case FAMILIA_INFORMATICA:
                        System.out.print("\n\tSist. m. i xarxes
(111)\n"+
n"+
n"+
                        "\tDesenv. D'ap. multiplataforma (121)\n"
                        "\tAdm. de sistemes informàtics (122)\n"
                        "\tDesenv. d'aplicacions web (123)\n");
                        break;
                    case FAMILIA_ADMINISTRACIO:
                        System.out.print("\n\tGestió adm. (211)\n"+
                        "\tAdministració i finances (221)\n"+
                        "\tAssitència a la direcció (222)\n");
                        break;
                    //Resta de families per definir
                }
            }
            do{
                System.out.print("\nEntra el cicle:");
                dadaCorrecta = scan.hasNextInt();
                if ( dadaCorrecta ){
                    cicle[i] = scan.nextInt();
                    switch (familia[i]){
                        case FAMILIA_INFORMATICA:
                            switch (cicle[i]){
                                case SMX: case DAM: case ASX:
case DAW:
                                break; //Si codi correcte no
                                fem res
                                default:
                                    dadaCorrecta = false;
                            }
                        }
                    }
                    break;
                }
            } while (!dadaCorrecta);
            case FAMILIA_ADMINISTRACIO:
                switch (cicle[i]){

```

```

fem res
        case GAD: case AFI: case ADI:
            break; //Si codi correcte no

        default:
            dadaCorrecta = false;
        }
        break;
        //Resta de families per definir
    }
    }
    intents++;
} while ((!dadaCorrecta)&&(intents<MAX_INTENTS));
}
intents++;
} while ((!dadaCorrecta)&&(intents<MAX_INTENTS));
}

if(dadaCorrecta){ //si fins al moment tot correcte, anem a llegir
nota
    intents=0; //Inicialitzem el comptador d'intents a 0
    do{
        System.out.print("Entra la nota mitjana: ");
        dadaCorrecta = scan.hasNextFloat();//per analitzar tipus de
dada
        if ( dadaCorrecta ){ // si és un número real
            nota[i] = scan.nextFloat();
            if((nota[i] < MINIM_NOTA)|| (nota[i] > MAXIM_NOTA)){
                dadaCorrecta = false;
            }
        }else{
            scan.next(); //buidem el buffer d'entrada per si entren
            // incorrectament un valor amb decimals.
        }
        intents++;
    } while ((!dadaCorrecta)&&(intents<MAX_INTENTS));
}

if(dadaCorrecta){
    i++; //index. També ens indica el número alumnes
}else{
    System.out.println("dades incorrectes");
}

if(i<MAX_ALUMNES){
    //Avaluem si vol introduir més alumnes
    System.out.print("es vol introduir mes alumnes?(si:1/no:0) ");
    dadaCorrecta = scan.hasNextInt();
    if ( dadaCorrecta ){ // si és un número sencer
        seguir = scan.nextInt();
        if(seguir == 1){
            introduirMes = true;
        }
    }else{
        scan.next();
        System.out.println("dada incorrecta");
    }
}
}while((introduirMes)&&(i<MAX_ALUMNES));

//Mostrem dades introduïdes des de 0 fins a i. (i conté el número
//d'alumnes entrats.
System.out.println("S'han inscrit "+i+" nous alumnes");
System.out.println("DNI      \tnota\tCicle");

```

```

for(int j=0;j<i;j++){
    //Obtinc nom del cicle
    switch (cicle[j]){
        case SMX:
            nomCicle = SMX_NOM;
            break;
        case DAM:
            nomCicle = DAM_NOM;
            break;
        case ASX:
            nomCicle = ASX_NOM;
            break;
        case DAW:
            nomCicle = DAW_NOM;
            break;
        case GAD:
            nomCicle = GAD_NOM;
            break;
        case AFI:
            nomCicle = AFI_NOM;
            break;
        case ADI:
            nomCicle = ADI_NOM;
            break;
        //Resta de families per definir
    }
    System.out.println(dni[j]+"\\t"+nota[j]+"\\t"+nomCicle);
}

//Endrecem
//Avaluem si es vol endreçar
System.out.print("vols mostrar els alumnes endreçats per
nota?(si:1/no:0) ");
dadaCorrecta = scan.hasNextInt();
if ( dadaCorrecta ){ // si és un número sencer
    seguir = scan.nextInt();
    if(seguir == 1){ //si volem endreçar
        //Ordenació -----
        for (int k = 0; k < i - 1; k++) {
            //Bucle intern.
            //Se cerca entre la resta de posicions quin és el valor més
            baix.
            for(int j = k + 1; j < i; j++) {
                //La posició tractada té un valor més alt que el de la
                cerca.
                // Els intercanviem.
                if (nota[k] > nota[j]) {
                    //Per intercanviar valors cal una variable auxiliar.
                    fcanvi = nota[k];
                    nota[k] = nota[j];
                    nota[j] = fcanvi;
                    //intercanviem resta de valors de la mateixa manera
                    //DNI
                    icanvi = dni[k];
                    dni[k] = dni[j];
                    dni[j] = icanvi;
                    //familia
                    icanvi = familia[k];
                    familia[k] = familia[j];
                    familia[j] = icanvi;
                    //cicle
                    icanvi = cicle[k];
                    cicle[k] = cicle[j];
                    cicle[j] = icanvi;
                }
            }
        }
    }
}

```

```

    }
}
//Fi ordenació -----
//Mostrem ordenats -----
System.out.println("DNI      \tnota\tCicle");
for(int j=0;j<i;j++){
    //Obtinc nom del cicle
    switch (cicle[j]){
        case SMX:
            nomCicle = SMX_NOM;
            break;
        case DAM:
            nomCicle = DAM_NOM;
            break;
        case ASX:
            nomCicle = ASX_NOM;
            break;
        case DAW:
            nomCicle = DAW_NOM;
            break;
        case GAD:
            nomCicle = GAD_NOM;
            break;
        case AFI:
            nomCicle = AFI_NOM;
            break;
        case ADI:
            nomCicle = ADI_NOM;
            break;
        //Resta de families per definir
    }
    System.out.println(dni[j]+" \t "+nota[j]+" \t "+nomCicle);
}
//Fi mostrar -----
}
}else{
    System.out.println("dada incorrecta");
}
}
}

```