

PROGRAMES BÀSICS

C#

ISAAC GUISSET SÁNCHEZ
DAW1 (subgrup A)

INTRODUCCIÓ

La finalitat d'aquest document és relatar els diferents aprenentatges que he anat adquirint a mesura de fer els exercicis d'iniciació i alhora demostrar que aquests han estat fets per mi i no per una intel·ligència artificial...

- Cada exercici tindrà el seu apartat, que correspondrà al seu mateix número.
ex1 = E1 ...
- A cada apartat hi haurà tres parts
1a part = 1P ...
 - 1P - de manera escrita explico l'estructura que tindrà el meu programa però sense entrar en detall, és dir com a plantejament inicial.
 - 2P – aquesta ja es tractarà d'un text que ara sí, de manera més tècnica expliqui les coses que he anat fent i per últim el programa complert (una captura o text formatat perquè es vegi com a codi).
 - 3P – en aquesta ja podrem veure com s'executa el programa sense inconvenients ni inconsistències al codi.

DESENVOLUPAMENT DE LA PRÀCTICA

E1 (Enunciat)

Un programa que demani el nom i el cognom de l'usuari i que el presenti.

P1 (Esquema inicial)

- 1- Imprimir un text per la pantalla que li demani a l'usuari el nom
- 2- Rebre el que l'usuari introdueix
- 3 – Imprimir un text per la pantalla que li demani a l'usuari el cognom
- 4 – Rebre el que l'usuari introdueix
- 5 – Retornar a l'usuari un text amb el nom i el cognom que hem guardat.

P2 (Esquema tècnic i codi)

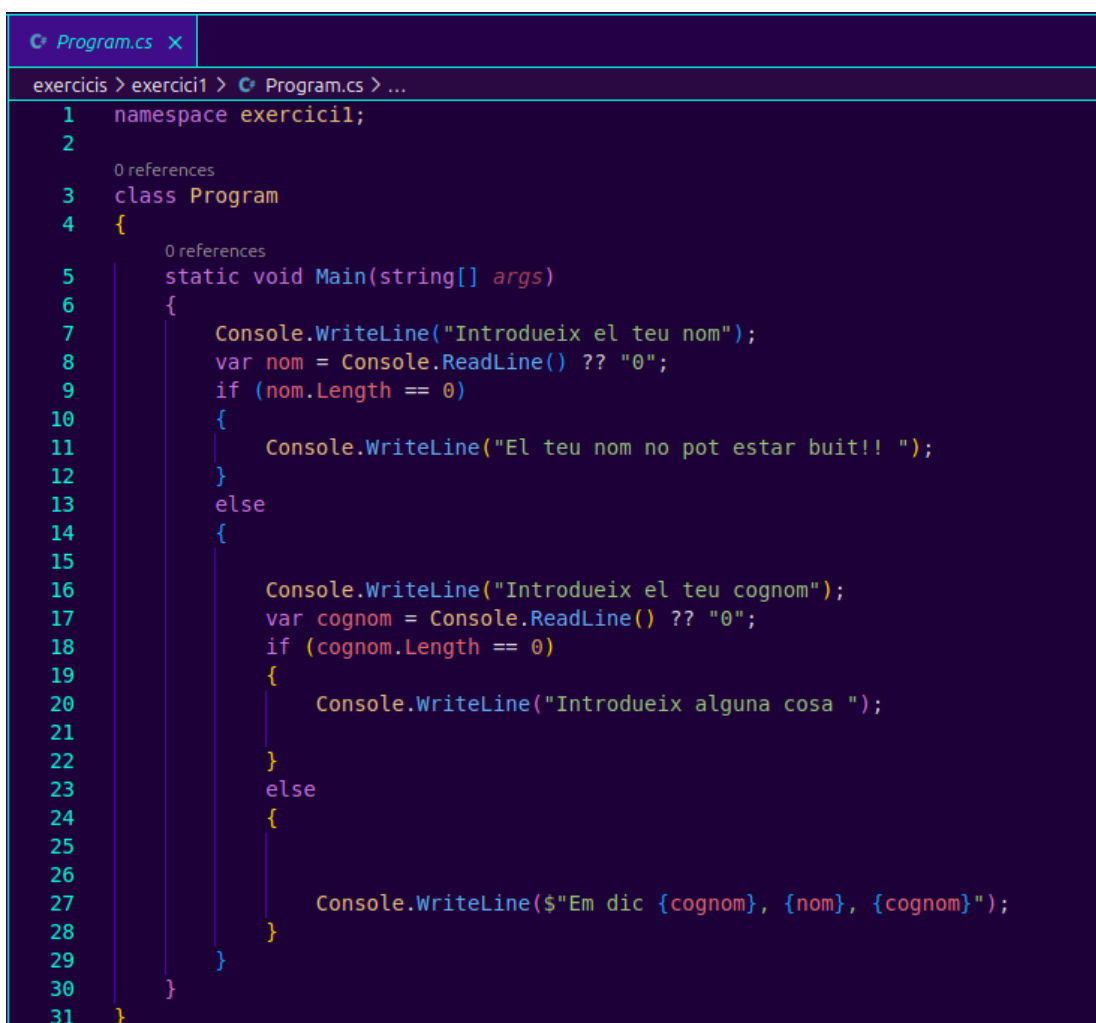
- 1- Imprimir un text per la pantalla que li demani a l'usuari el nom
Assignem el valor que l'usuari introdueixi a la variable nom, que serà un string ja que tot el que entra a partir d'un readline es string
`Console.WriteLine("Introdueix el teu nom");`
- 2- Rebre el que l'usuari introdueix.
`var nom = Console.ReadLine();`
- 3 – Imprimir un text per la pantalla que li demani a l'usuari el cognom
`Console.WriteLine("Introdueix el teu cognom");`
- 4 – Rebre el que l'usuari introdueix
Assignem el valor que l'usuari introdueixi a la variable cognom, que serà un string ja que tot el que entra a partir d'un readline es string
`var cognom = Console.ReadLine();`
- 5 – Retornar a l'usuari un text amb el nom i el cognom que hem guardat.
Utilitzem \$ davant de les cometes dobles per inserir el contingut per tal de poder introduir les variables i que s'imprimeixi el seu valor a la terminal conjunt amb el meu text
`Console.WriteLine($"El meu nom és {cognom}, {nom}, {cognom}")`

6 – Afegir una petita condició amb if-else per tal de que si el nom o el cognom estan buits ens faci un print de El teu nom/cognom no pot estar buit

He afegit dues condicions if else, enganxades.

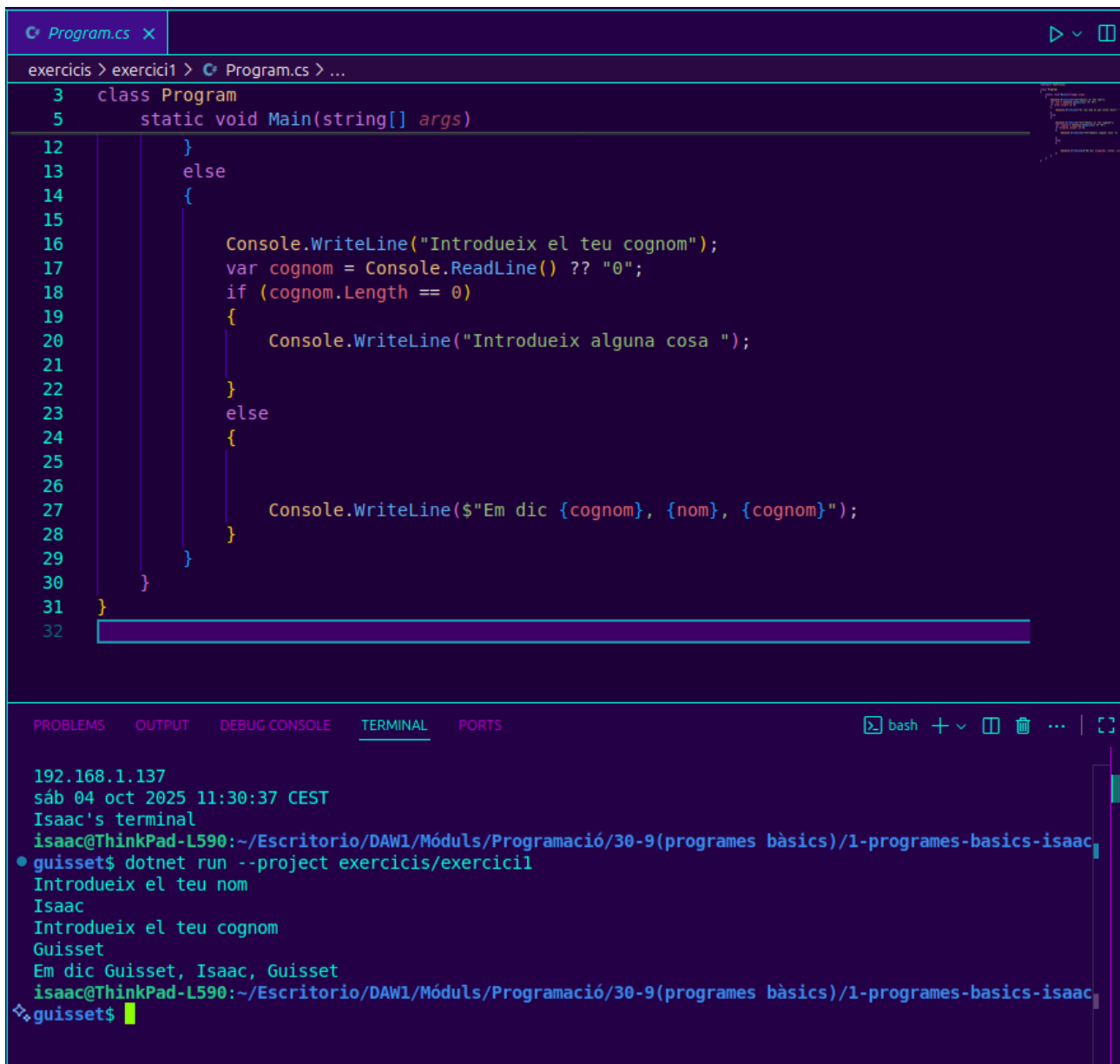
El primer if diu que si `nom.Length == 0` faci un print de que el teu nom no pot estar buit amb `consolewriteline`, si aquesta condició no es compleix, segueix el programa normalment preguntant el cognom, i un cop introduït aquest torna a fer la comprovació `if cognom.Length == 0`... si les dues condicions no es compleixen passa directament al else, que ja és la sortida final del programa.

Encara no domino del tot els bucles, però amb un bucle podem estalviar-nos haver de re-executar el programa cada vegada que falli



```
1 namespace exercicil;
2
3 class Program
4 {
5     static void Main(string[] args)
6     {
7         Console.WriteLine("Introdueix el teu nom");
8         var nom = Console.ReadLine() ?? "0";
9         if (nom.Length == 0)
10        {
11            Console.WriteLine("El teu nom no pot estar buit!! ");
12        }
13        else
14        {
15            Console.WriteLine("Introdueix el teu cognom");
16            var cognom = Console.ReadLine() ?? "0";
17            if (cognom.Length == 0)
18            {
19                Console.WriteLine("Introdueix alguna cosa ");
20            }
21            else
22            {
23                Console.WriteLine($"Em dic {cognom}, {nom}, {cognom}");
24            }
25        }
26    }
27 }
28
29
30
31 }
```

P3 (Execució correcta del codi i demostració de l'ús de if-else)



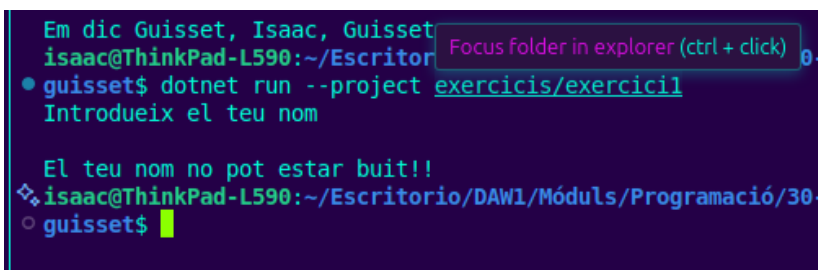
The screenshot shows a Visual Studio Code editor with a C# file named `Program.cs`. The code defines a `class Program` with a `Main` method. The `Main` method prompts the user to enter their name and surname. It uses `Console.WriteLine` and `Console.ReadLine` to interact with the user. An `if-else` statement checks if the surname is empty. If it is, it prompts the user to enter a surname. Otherwise, it prints the full name.

```
3 class Program
5     static void Main(string[] args)
6     {
7         Console.WriteLine("Introdueix el teu nom");
8         var nom = Console.ReadLine() ?? "";
9         Console.WriteLine("Introdueix el teu cognom");
10        var cognom = Console.ReadLine() ?? "";
11        if (cognom.Length == 0)
12        {
13            Console.WriteLine("Introdueix alguna cosa ");
14        }
15        else
16        {
17            Console.WriteLine($"Em dic {nom}, {cognom}");
18        }
19    }
20 }
```

The terminal output shows the execution of the program. The user enters "Isaac" for the name and "Guisset" for the surname. The program outputs "Em dic Guisset, Isaac, Guisset".

```
192.168.1.137
sáb 04 oct 2025 11:30:37 CEST
Isaac's terminal
isaac@ThinkPad-L590:~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-programes-basics-isaac
guisset$ dotnet run --project exercicis/exercici1
Introdueix el teu nom
Isaac
Introdueix el teu cognom
Guisset
Em dic Guisset, Isaac, Guisset
isaac@ThinkPad-L590:~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-programes-basics-isaac
guisset$
```

si no introdueix res a nom o a cognom:



The screenshot shows a terminal window where the user enters an empty name. The program outputs "El teu nom no pot estar buit!!".

```
Em dic Guisset, Isaac, Guisset
isaac@ThinkPad-L590:~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-programes-basics-isaac
guisset$ dotnet run --project exercicis/exercici1
Introdueix el teu nom

El teu nom no pot estar buit!!
isaac@ThinkPad-L590:~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-programes-basics-isaac
guisset$
```

E2 (Enunciat)

Declara variables per emmagatzemar l'adreça d'una persona: carrer, número, codi postal i població. Imprimeix el resultat en una sola línia.

En aquest programa he entès que les variables s'han de crear sense “ajuda” de l'usuari, simulant que les dades son en una base de dades, pero podriem fer el mateix amb ajuda de l'usuari utilitzant Console.ReadLine();

P1 (Esquema inicial)

1-Declarar les dades perquè es guardin a la memòria

2-Un cop emmagatzemades, retornarles en una sola fila imprimint per la terminal

P2 (Esquema tècnic i codi)

1-Declarar les variables perquè es guardin a la memòria

var carrer = “narcis munturiol”;

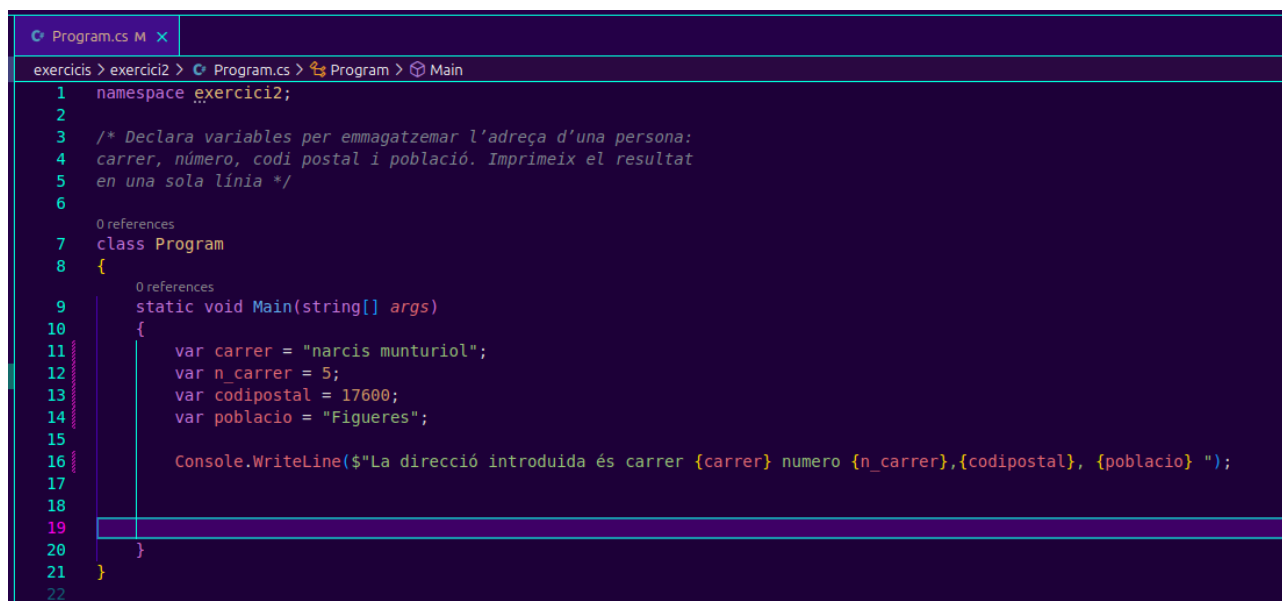
var n_carrer = 5;

var codipostal = 17600;

var poblacio = “Figueres”;

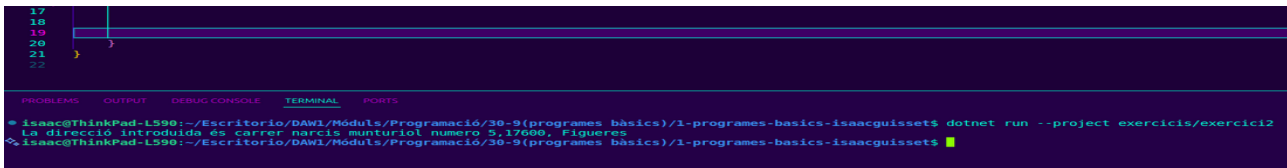
2-Un cop emmagatzemades, retornarles en una sola fila imprimint per la terminal

Utilitzarem el Console.WriteLine(“La direcció introduïda és {carrer}...”)



```
1 namespace exercici2;
2
3 /* Declara variables per emmagatzemar l'adreça d'una persona:
4 carrer, número, codi postal i població. Imprimeix el resultat
5 en una sola línia */
6
7 class Program
8 {
9     static void Main(string[] args)
10    {
11        var carrer = "narcis munturiol";
12        var n_carrer = 5;
13        var codipostal = 17600;
14        var poblacio = "Figueres";
15
16        Console.WriteLine($"La direcció introduïda és carrer {carrer} numero {n_carrer},{codipostal}, {poblacio} ");
17    }
18 }
19
20
21
22
```

P3 (Execució correcta del codi)



The screenshot shows a code editor with a C# program. The code defines a class `Producte` with properties `nom`, `preu`, `estoc`, and `compres_ultim_any`. It includes a `ToString` method that returns a formatted string. The terminal output shows the command `dotnet run --project exercicis/exercici2` being executed, and the output of the program, which prints the details of a product named "Ratolí ThunderX3" with a price of 35.99, currently in stock, and first purchased in 300.

E3 (Enunciat)

Defineix quatre variables per al nom d'un producte, el seu preu i si esta en estoc o no. Imprimeix la informació de forma clara

P1 (Esquema inicial)

- 1- Definir les variables de cada camp que necessitem
- 2- Crear una variable per definir que hi ha estoc o no de manera més fàcil que `true.false`
- 3-Retornar el contingut junt a la terminal

P2 (Esquema tècnic i codi)

- 1-Definir les variables de cada camp que necessitem

```
var nom = "Ratolí ThunderX3";  
double preu = 35.99;  
bool estoc = false;  
int compres_ultim_any = 300;
```

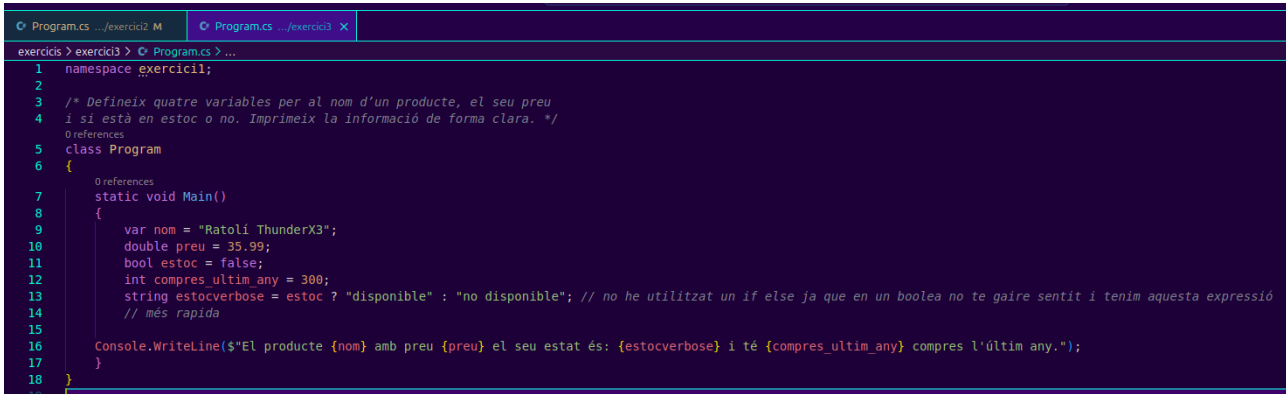
- 2-Crear una variable per definir que hi ha estoc o no de manera verbose

```
string estocverbose = estoc ? "disponible" : "no disponible"; // no he utilitzat un  
if else ja que en un boolea no te gaire sentit i tenim aquesta expressió  
// més rapida
```

la sintaxis es `estoc ? Variables>true : variables>false;`

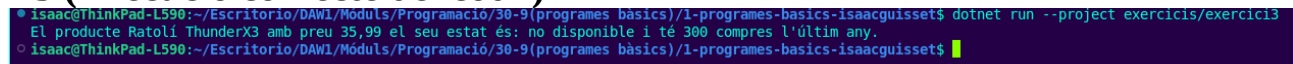
3-Retornar el contingut junt a la terminal

Utilitzem `Console.WriteLine($"EL producte {nom } amb preu")`



```
1 namespace exercicis;
2
3 /* Defineix quatre variables per al nom d'un producte, el seu preu
4 i si està en estoc o no. Imprimeix la informació de forma clara. */
5 class Program
6 {
7     static void Main()
8     {
9         var nom = "Ratoli ThunderX3";
10        double preu = 35.99;
11        bool estoc = false;
12        int compres_ultim_any = 300;
13        string estocverbose = estoc ? "disponible" : "no disponible"; // no he utilitzat un if else ja que en un boolea no te gaire sentit i tenim aquesta expressió
14        // més rapida
15
16        Console.WriteLine($"El producte {nom} amb preu {preu} el seu estat és: {estocverbose} i té {compres_ultim_any} compres l'últim any.");
17    }
18 }
```

P3 (Execució correcta del codi)



```
isaac@ThinkPad-L590:~/Escritorio/DAW1/Moduls/Programacio/30-9(programes basics)/1-programes-basics-isaacguisset$ dotnet run --project exercicis/exercici3
El producte Ratoli ThunderX3 amb preu 35,99 el seu estat és: no disponible i té 300 compres l'últim any.
isaac@ThinkPad-L590:~/Escritorio/DAW1/Moduls/Programacio/30-9(programes basics)/1-programes-basics-isaacguisset$
```


E4 (Enunciat)

Un avi encara no sap quin és el valor de les coses perquè a tot arreu li pael d'euros però ell s'ha negat a canviar i encara té un pot ple de monedes de pesseta

Necessita un programa que li digui quin és el valor en les antigues pessetes

P1 (Esquema inicial)

1. Definir una variable amb el valor en euros que volem convertir.
2. Definir una constant amb el valor d'una pesseta respecte a l'euro (1 € = 166.386 pessetes).
3. Fer la conversió multiplicant el valor en euros per la constant.
4. Mostrar per pantalla el resultat de forma clara.

P2 (Esquema tècnic i codi)

1-Definir les variables necessàries

```
const double CONVERSIO = 166.386; es una constant ja que mai canvia, un euro sempre equivaldra a la mateixa quantitat de pesetes
```

```
Console.Write("Introdueix el valor en euros: ");  
string input = Console.ReadLine();
```

2-Validem l'entrada de l'usuari

```
if (double.TryParse(input, out double euros))
```

3- Fem la conversió i mostrem el resultat

```
int pessetes = (int)Math.Round(euros * CONVERSIO);  
Console.WriteLine($"{euros}€ equivalen a {pessetes} pessetes.");;
```

4- si hi hagues un error amb el valor en euros:

```
else  
{  
    Console.WriteLine("Has d'introduir un número vàlid.");  
}
```

Codi complet

```
namespace exercici4
{
    /* Un avi encara no sap quin és el valor de les coses perquè a tot arreu li parlen d'euros,
    però ell s'ha negat a canviar i encara té un pot ple de monedes de pesseta.
    Necessita un programa amb el qual posi el valor de l'article que vol comprar en
    euros i li digui quin és el valor en les antigues pessetes.
    Nota: 1 euro = 166.386 pessetes i les pessetes no tenien cap decimal. */

    0 references
    class Program
    {
        0 references
        static void Main()
        {
            const double CONVERSIÓ = 166.386;

            Console.WriteLine("Hola avi, introdueix el valor en euros: ");
            string input = Console.ReadLine();

            if (double.TryParse(input, out double euros)) // comprova que sigui un número valid
            {
                int pessetes = (int)Math.Round(euros * CONVERSIÓ); // les pessetes no tenen decimals per tant fem mathround
                Console.WriteLine($"{euros}€ equivalen a {pessetes} pessetes.");
            }
            else
            {
                Console.WriteLine("Has d'introduir un número vàlid.");
            }
        }
    }
}
```

P3 (Execució correcte del codi)

```
Has d'introduir un número vàlid.
isaac@ThinkPad-L590:~/Escritorio/DAW1/Móduls/Programació/30-9(programes bàsics)/1-programes-basics-isaacguisset$ dotnet run --project exercicis/exercici4
Hola avi, introdueix el valor en euros: 88
88€ equivalen a 14642 pessetes.
isaac@ThinkPad-L590:~/Escritorio/DAW1/Móduls/Programació/30-9(programes bàsics)/1-programes-basics-isaacguisset$
```

E5 (Enunciat)

Feu un programa que demani un nom i l'any de naixement i imprimeixi per pantalla

“Hola [nom]! Ja tens [edat] anys?

nom i edat seran el nom introduït i l'edat la resta de l'any actual i l'any de naixement.

P1 (Esquema inicial)

1- Demanar i recollir nom i any de naixement

2-Crear variable de l'any actual (2025)

3-Crear variable de l'edat en funció de l'any de naixement i l'any actual

P2 (Esquema tècnic i codi)

1- Utilitzem Console.WriteLine() per demanar el nom i l'any de naixement.

Llegim les respostes amb Console.ReadLine().

El nom el guardem en una variable string, i l'any de naixement el convertim a número amb Convert.ToInt16().

```
Console.WriteLine("Introdueix el teu nom");
```

```
var nom = Console.ReadLine();
```

```
Console.WriteLine("Introdueix el teu any de naixement");
```

```
var any_naix = Convert.ToInt16(Console.ReadLine());
```

2-Definir l'any actual i calcular l'edat

```
int anyactual = 2025;
```

```
var edat = anyactual - any_naix;
```

3-Mostrar el resultat per pantalla

Finalment mostrem un missatge amb l'edat i el nom

```
Console.WriteLine($"Hola {nom}! ja tens {edat} anys ");
```

P3 (Execució correcta del codi)

```
... 192.168.1.137
... sàb 04 oct 2025 12:43:17 CEST
... Isaac's terminal
... isaac@ThinkPad-L590:~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-
... programes-basics-isaacguisset$ dotnet run --project exercicis/exercici5
... Introdueix el teu nom
... Isaac
... Introdueix el teu any de naixament
... 2007
... Hola Isaac! ja tens 18 anys
... isaac@ThinkPad-L590:~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-
... programes-basics-isaacguisset$
```

E6 (Enunciat)

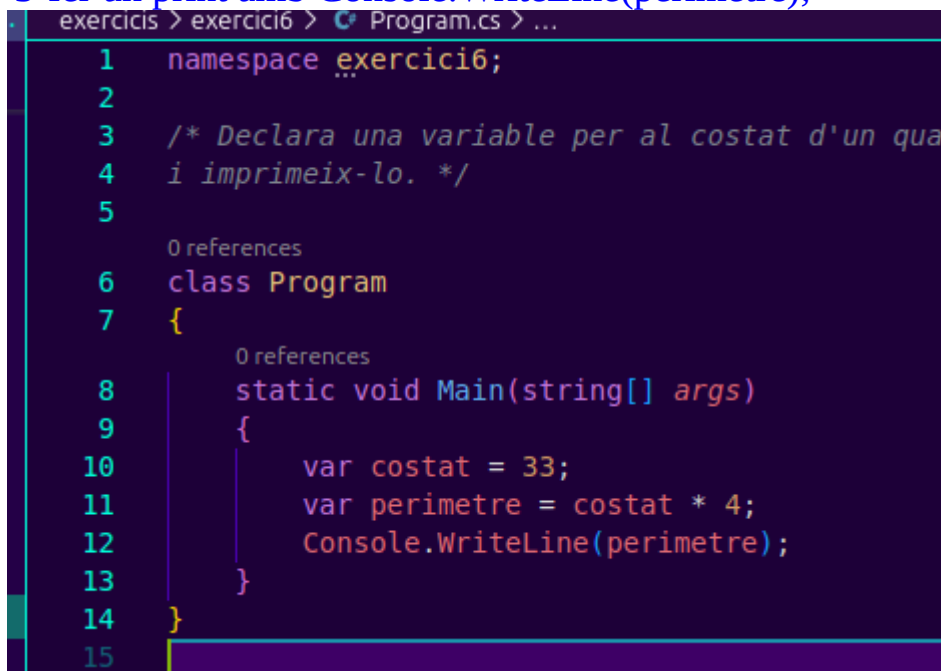
Declara una variable per al costat d'un quadrat i calcula el seu perímetre, després imprimeix-lo

P1 (Esquema inicial)

- 1-Definir la variable del costat del quadrat
- 2-Definir la variable del perímetre en funció del costat
- 3-retornar escrit el perímetre

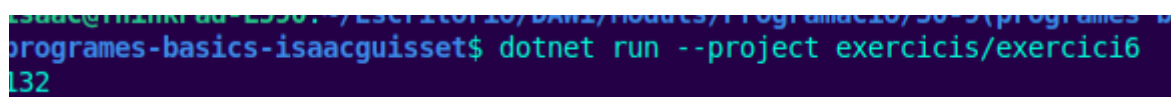
P2 (Esquema tècnic i codi)

- 1- definir la variable amb `var costat = 33;`
- 2-definir la variable amb `var perímetre = costat * 4`
- 3-fer un print amb `Console.WriteLine(perímetre);`



```
exercicis > exercici6 > Program.cs > ...
1 namespace exercici6;
2
3 /* Declara una variable per al costat d'un qua
4 i imprimeix-lo. */
5
6 0 references
7 class Program
8 {
9     0 references
10     static void Main(string[] args)
11     {
12         var costat = 33;
13         var perímetre = costat * 4;
14         Console.WriteLine(perímetre);
15     }
16 }
```

P3 (Execució correcte del codi)



```
isaac@thinkpad-L550: ~/Escritori/DAW1/Notes/Programació/30-5(programes b
programes-basics-isaacguisset$ dotnet run --project exercicis/exercici6
132
```

E7 (Enunciat)

Defineix tres variables amb tres notes numèriques.
Calcula la mitjana i mostra el resultat per pantalla.

P1 (Esquema inicial)

- 1- Definir les variables de les notes
- 2- Calcular la suma
- 3- Calcular la mitjana i arrodonir-la
- 4- Mostrar el resultat per pantalla

P2 (Esquema tècnic i codi)

- 1- Definir les variables amb les tres notes amb double

```
double n_matematiques = 5.50;  
double n_angles = 3.50;  
double n_basedades = 7.7;
```

- 2- Calcular la suma i la mitjana de les notes

es crea una variable amb suma que té la suma de les tres notes
despres es calcula la mitjana dividint aquesta suma entre 3

```
var suma = n_matematiques + n_angles + n_basedades;  
var mitjana = suma / 3;
```

- 3-Arrodonir la mitjana ja que la mitjana pot tenir molts decimals posarem
Math.Round()

```
var mitjana_dosdecimals = Math.Round(mitjana, 2);
```

- 4- mostrar el resultat en pantalla

```
Console.WriteLine($"La mitjana és {mitjana_dosdecimals}");
```

```
1 namespace exercici7;
2
3 /* Defineix tres variables amb tres notes numèriques.
4    Calcula la mitjana i mostra el resultat per pantalla. */
5
6 class Program
7 {
8     static void Main(string[] args)
9     {
10         double n_matematiques = 5.50;
11         double n_angles = 3.50;
12         double n_basedades = 7.7;
13         var suma = n_matematiques + n_angles + n_basedades;
14         var mitjana = suma / 3;
15         var mitjana_dosdecimals = Math.Round(mitjana, 2);
16         Console.WriteLine($"La mitjana és {mitjana_dosdecimals}");
17     }
18 }
19
```

P3 (Execució correcte del codi)

```
isaac@ThinkPad-L590: ~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)
programes-basics-isaacguisset$ dotnet run --project exercicis/exercici7
La mitjana és 5,57
isaac@ThinkPad-L590:~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)
programes-basics-isaacguisset$
```

E8 (Enunciat)

Assigna un nombre total de minuts a una variable.
Calcula quantes hores senceres i minuts restants són i imprimeix el resultat.

P1 (Esquema inicial)

- 1- Assigna un nombre total de minuts
- 2- Defineix quants minuts té una hora
- 3- Es calcula quantes hores senceres hi ha dins dels minuts restants i es calcula quants minuts sobren .
- 4- mostrem per pantalla el resultat en format

P2 (Esquema tècnic i codi)

- 1- Declarem les variable minuts i hores_minut

```
var minuts = 120;  
var hores_minut = 60;
```

- 2- declarem la variable hores com a resultat de minuts dividit per hores minut

```
var hores = minuts / hores_minut;
```

- 3- declarem la variable minutsrestants com a resultat de els minuts amb mòdul de les hores minut osigui 60.

```
var minutsrestants = minuts % hores_minut;
```

CODI:

```
1 namespace exercici8;  
2  
3 /* Assigna un nombre total de minuts a una variable.  
4 Calcula quantes hores senceres i minuts restants són i imprimeix el resultat. */  
5  
6 0 references  
7 class Program  
8 {  
9     0 references  
10    static void Main(string[] args)  
11    {  
12        var minuts = 120;  
13        var hores_minut = 60;  
14        var hores = minuts / hores_minut;  
15        var minutsrestants = minuts % hores_minut;  
16        Console.WriteLine($"{hores} hores i {minutsrestants} minuts");  
17    }  
18 }
```


P3 (Execució correcta del codi)

```
isaac@ThinkPad-L590:~/Escritorio/DAW1/Móduls/Programació/30-9(programes bàsics)/1-programes-basics-1  
guisset$ dotnet run --project exercicis/exercici8  
2 hores i 0 minuts  
isaac@ThinkPad-L590:~/Escritorio/DAW1/Móduls/Programació/30-9(programes bàsics)/1-programes-basics-1
```

E9 (Enunciat)

Demana un valor en metres. Converteix el valor a peus (1 metre = 3,28084 peus) i imprimeix el resultat.

P1 (Esquema inicial)

- 1-Demana un valor en metres i el guarda
- 2-Una constant de la conversió entre metres i peus
- 3-A través de la constant calculem la conversió a peus multiplicant
- 4- Mostrem per pantalla el resultat

P2 (Esquema tècnic i codi)

- 1- Mostrem el missatge a l'usuari per demanar que introdueixi el valor en metres

`Console.WriteLine("Introdueix el valor en metres per tal de convertir-lo a peus");`

- 2- Amb `var metres = Convert.ToInt32(Console.ReadLine());` llegim el que introdueix l'usuari, el convertim a int i fem que la variable metres correspongui a aquest int

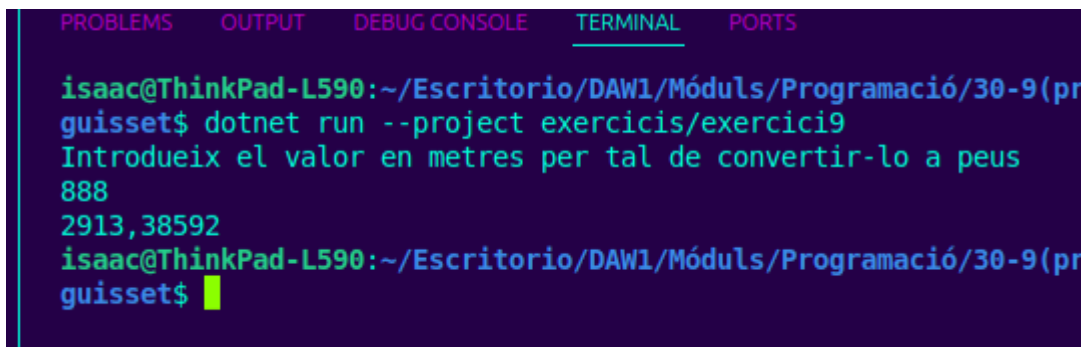
- 3- Defineixo la constant de conversió de `double peu = 3.28084;`

- 4- Calculem el resultat de la conversió multiplicant els metres pel valor d'un peu amb `var resultat = metres * peu;`

- 5- Mostrem el resultat final per pantalla amb `Console.WriteLine(resultat);`

```
1 namespace exercici9;
2
3 /* Demana un valor en metres. Converteix el valor a peus (1 metre = 3,28084 peus)
4 i imprimeix el resultat. */
5
6 0 references
7 class Program
8 {
9     0 references
10     static void Main(string[] args)
11     {
12         Console.WriteLine("Introdueix el valor en metres per tal de convertir-lo a peus");
13         var metres = Convert.ToInt32(Console.ReadLine());
14         double peu = 3.28084;
15         var resultat = metres * peu;
16         Console.WriteLine(resultat);
17     }
18 }
```

P3 (Execució correcte del codi)



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

isaac@ThinkPad-L590:~/Escritorio/DAW1/Móduls/Programació/30-9(pr
guisset$ dotnet run --project exercicis/exercici9
Introdueix el valor en metres per tal de convertir-lo a peus
888
2913,38592
isaac@ThinkPad-L590:~/Escritorio/DAW1/Móduls/Programació/30-9(pr
guisset$
```

E10 (Enunciat)

Declara dues variables: una per al nom i una altra per l'any de naixement. Combina-les per crear una contrasenya bàsica (per exemple, Pere i 1982 -> Pere1982).

Imprimeix la contrasenya generada (i sobretot no la facis servir mai que és molt dolenta)

P1 (Esquema inicial)

- 1- Definim valor de nom i de any de naixament
- 2-Definim el valor de contrassenya amb nom i anynaix
- 3-Retornem amb un print el valor de la variable contrassenya

P2 (Esquema tècnic i codi)

- 1- definim les dues variables amb `var nomvariable = valor;`
- 2- definim la variable contrassenya concatenant les dues amb un `+`
`var contrassenya = nom + anynaix;`
- 3-fem un print amb `Console.WriteLine(contrassenya);`

codi:

```
exercicis > exercici10 > Program.cs > ...
1 namespace exercici10;
2
3 /* Declara dues variables: una per al nom i una altra
4 Combina-les per crear una contrasenya bàsica (per exen
5 Imprimeix la contrasenya generada (i sobretot no la fa
6
7 0 references
8 class Program
9 {
10     0 references
11     static void Main(string[] args)
12     {
13         var nom = "Isaac";
14         var anynaix = 2007;
15         var contrassenya = nom + anynaix;
16         Console.WriteLine(contrassenya);
17     }
18 }
```

P3 (Execució correcta del codi)

```
programes-basics-isaacguisset$ dotnet run --project exercicis/exercici10
Isaac2007
isaac@ThinkPad-L590:~/Escritorio/DAW1/Móduls/Programació/30-9(programes b
programes-basics-isaacguisset$
```

E11 (Enunciat)

A partir de dues variables (nom d'usuari i domini), crea i imprimeix una adreça de correu electrònic completa.

P1 (Esquema inicial)

- 1- Demanar el nom d'usuari i el domini & guardar l'informació
- 2-Generar el correu concatenant el nom d'usuari, el domini i l'arroba
- 3- retornar a l'usuari el correu sencer

P2 (Esquema tècnic i codi)

- 1-Demanar el nom d'usuari i el domini amb un `Console.WriteLine()`;
- 2-Guardar el que diu l'usuari a través d'un `Console.ReadLine()`;
`var nom_usuari = Console.ReadLine()` i `var domini = Console.ReadLine()`
- 3- Concatenar les dues variables emmagatzemades i al mig posar-hi un `@`
`var correu = nom_usuari + "@" + domini;`
- 4- Retornar el resultat de la concatenació per consola
`Console.WriteLine(correu);`

```
1 namespace exercicill;
2
3 // A partir de dues variables (nom d'usuari i domini), crea i imprimeix una adreça
  0 references
4 class Program
5 {
  0 references
6     static void Main(string[] args)
7     {
8         Console.WriteLine("Introdueix el teu nom d'usuari: ");
9         var nom_usuari = Console.ReadLine();
10        Console.WriteLine("Introdueix el teu domini: ");
11        var domini = Console.ReadLine();
12        var correu = nom_usuari + "@" + domini;
13        Console.WriteLine($"La teva direcció de correu és {correu}");
14    }
15 }
16
17
```

P3 (Execució correcta del codi)

```
isaac@ThinkPad-L590:~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-programes-basics-isaac
guisset$ dotnet run --project exercicis/exercicill
Introdueix el teu nom d'usuari:
iguissetsanchez
Introdueix el teu domini:
cendrassos.net
La teva direcció de correu és iguissetsanchez@cendrassos.net
isaac@ThinkPad-L590:~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-programes-basics-isaac
guisset$
```

E11 (Enunciat)

Fes un programa que a partir de dos números imprimeixi per pantalla la seva suma, la seva resta, la seva multiplicació i la seva divisió

Entra el primer número: 3

Entra el segon número: 4

$$4 + 3 = 7$$

$$4 - 3 = 1$$

$$4 * 5 = 12$$

$$4 / 3 = 1 \text{ i en sobra } 1$$

Entra el primer número: 25

Entra el segon número: 5

$$25 + 5 = 30$$

$$25 - 5 = 20$$

$$25 * 5 = 125$$

$$25 / 5 = 5 \text{ i en sobra } 0$$

P1 (Esquema inicial)

- 1- Mostrem un missatge explicant què fa el programa.
- 2- Demanem a l'usuari que introdueixi el primer número.
- 3- Llegim el valor introduït i el convertim a enter.
- 4- Demanem a l'usuari que introdueixi el segon número.
- 5- Llegim el segon valor i el convertim a enter.
- 6- Calculem la suma, la resta, la multiplicació, la divisió i el residu de la divisió dels dos nombres.
- 7- Mostrem tots els resultats per pantalla.
- 8- Si l'usuari introdueix un valor no vàlid (com text en lloc d'un número), capturem l'error amb trycatch i ensenyanyem un missatge d'error.

P2 (Esquema tècnic i codi)

- 1- mostrem el missatge principal que explica la funcionalitat del programa
`Console.WriteLine("Aquest programa a partir de dos números retorna la seva suma, la seva resta, la seva multiplicació i la seva divisió");`
- 2- demanem el primer número a l'usuari
`Console.WriteLine("Introdueix el primer número");`
- 3- llegim i convertim a enter el primer número dins del try
`var a = Convert.ToInt32(Console.ReadLine());`
- 4- demanem el segon número a l'usuari
`Console.WriteLine("Introdueix el segon número");`
- 5- llegim i convertim a enter el segon número
`var b = Convert.ToInt32(Console.ReadLine());`
- 6- calculem les operacions bàsiques
`var suma = a + b;`
`var resta = a - b;`
`var multiplicacio = a * b;`
`var divisio = a / b;`
`var divisio_resta = a % b;`
- 7- mostrem els resultats amb un print
`Console.WriteLine($"{a} + {b} = {suma}");`
`Console.WriteLine($"{a} - {b} = {resta}");`
`Console.WriteLine($"{a} * {b} = {multiplicacio}");`
`Console.WriteLine($"{a} / {b} = {divisio} i en sobra {divisio_resta}");`

8- si l'usuari introdueix un valor no vàlid, el catch mostra un missatge d'error
catch (Exception)
Console.WriteLine("Aixó que has introduït no és un número");

P3 (Execució correcta del codi)

```
isaac@ThinkPad-L590:~/Escritorio/DAW1/Móduls/Programació/30-9(programes bàsics)/1-programes-basics-isaac-guisset$ dotnet run --project exercicis/exercici12
Aquest programa a partir de dos números retorna la seva suma, la seva resta, la seva multiplicació i la seva divisió
Introdueix el primer número
9
Introdueix el segon número
8
9 + 8 = 17
9 - 8 = 1
9 * 8 = 72
9 / 8 = 1 i en sobren 1
isaac@ThinkPad-L590:~/Escritorio/DAW1/Móduls/Programació/30-9(programes bàsics)/1-programes-basics-isaac-guisset$
```

E13 (Enunciat)

Un alumne ha inventat un sistema per codificar dates que consisteix a eliminar els separadors de les dates per fer no es vegi que són dates (la data 24/09/2024 la converteix en 24092024). El problema és que ho ha aplicat a tots els documents de l'institut i ara la direcció no sap ordenar els documents.

Necessitem un programa que faci el procés invers

Entra la data sense formatar: 25092024
La data és 25/09/2024

Entra la data sense formatar: 12031970
La data és 12/03/1970

P1 (Esquema inicial)

- 1- mostrem un missatge demanant a l'usuari que entri una data sense format
- 2- llegim el valor que introdueix l'usuari i el guardem a una variable anomenada data
- 3- comprovem si la llargada de la data és superior a 8 caràcters
- 4- si és més gran que 8, mostrem un missatge d'error dient que la data no és correcta
- 5- si la llargada és correcta (8 caràcters), separem el dia, el mes i l'any amb substring
- 6- mostrem la data formatejada en pantalla amb el format que toca

P2 (Esquema tècnic i codi)

- 1- mostrem el missatge per demanar a l'usuari la data sense formatar

```
Console.WriteLine("Entra la data sense formatar (DDMMMAAAA): ");
```

- 2- llegim la dada introduïda i la guardem en una variable

```
string data = Console.ReadLine();
```

- 3- comprovem si la llargada de la cadena és més gran que 8

```
if (data.Length > 8)
```

- 4- si la condició és certa, mostrem un missatge d'error

```
Console.WriteLine("La data que has introduït no és correcta, ha de tenir 8 caràcters i en format DDMMMAAAA - Exemple: 04202025");
```

- 5- si la condició no es compleix, extraïem les parts de la data

```
string dia = data.Substring(0, 2);
```

```
string mes = data.Substring(2, 2);
```

```
string any = data.Substring(4);
```

- 6- mostrem la data formatejada per pantalla

```
Console.WriteLine($"La data és {dia}/{mes}/{any}");
```



```

1 namespace Exercicis;
2
3 /* Un alumne ha inventat un sistema per codificar dates que consisteix a eliminar els separadors de les dates
4 per fer no es vegi que són dates (la data 24/09/2024 la converteix en 24092024).
5 El problema és que ho ha aplicat a tots els documents de l'institut i ara la direcció no sap
6 ordenar els documents.
7
8 Necessitem un programa que faci el procés invers
9
10 Entra la data sense formatar: 25092024
11 La data és 25/09/2024
12
13 Entra la data sense formatar: 12031970
14 La data és 12/03/1970 */
15
16 0 references
17 class Program
18 {
19     0 references
20     static void Main(string[] args)
21     {
22
23         Console.Write("Entra la data sense formatar (DDMMAAAA): ");
24
25         string data = Console.ReadLine();
26
27         if (data.Length > 8)
28         {
29             Console.WriteLine("La data que has introduït no és correcta, ha de tenir 8 caràcters i en format DDMMAAAA - Exemple: 04202025");
30         }
31         else
32         {
33             string dia = data.Substring(0, 2);
34             string mes = data.Substring(2, 2);
35             string any = data.Substring(4);
36             Console.WriteLine($"La data és {dia}/{mes}/{any}");
37         }
38     }
39 }
40
41
42
43
44

```

P3 (Execució correcta del codi)

```

isaac@ThinkPad-L590: ~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-programes-basics-isaacguisset$ dotnet run --project exercicis/exercici13
/home/isaac/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-programes-basics-isaacguisset/exercicis/exercici13/Program.cs(24,23): warning CS8600: Se va a convertir un literal nulo o un posib
le valor nulo en un tipo que no acepta valores NULL.
/home/isaac/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-programes-basics-isaacguisset/exercicis/exercici13/Program.cs(27,17): warning CS8602: Desreferencia de una referencia posiblemente
NULL.
Entra la data sense formatar (DDMMAAAA): 07102007
La data és 07/10/2007
isaac@ThinkPad-L590: ~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-programes-basics-isaacguisset$

```

en cas d'introduir malament el numero

```

La data és 07/10/2007
isaac@ThinkPad-L590: ~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-programes-basics-isaacguisset$ dotnet run --project exercicis/exercici13
Entra la data sense formatar (DDMMAAAA): 898989898
La data que has introduït no és correcta, ha de tenir 8 caràcters i en format DDMMAAAA - Exemple: 04202025
isaac@ThinkPad-L590: ~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-programes-basics-isaacguisset$

```

E14 (Enunciat)

Declara una paraula i fes que el programa imprimeixi només la primera lletra, l'última lletra i la lletra del mig

P1 (Esquema inicial)

- 1- declarem una variable amb una paraula assignada
- 2- agafem la primera lletra de la paraula
- 3- agafem la lletra del mig de la paraula
- 4- agafem l'última lletra de la paraula
- 5- mostrem per pantalla la primera lletra, la del mig i l'última

P2 (Esquema tècnic i codi)

- 1- declarem la variable amb la paraula

```
var paraula = "pastanaga";
```

- 2- guardem el primer caràcter de la paraula

```
char primercaracter = paraula[0];
```

- 3- obtenim el caràcter del mig dividint la longitud de la paraula entre 2

```
char caracterdelmig = paraula[paraula.Length / 2];
```

- 4- obtenim l'últim caràcter restant 1 a la longitud total

```
char caracterfinal = paraula[paraula.Length - 1];
```

- 5- mostrem tots els valors per pantalla

```
Console.WriteLine($"El primer caràcter és {primercaracter}, el caràcter del mig és {caracterdelmig} i el caràcter del final és {caracterfinal}");
```

codi

```
exercicis > exercici14 > C:\Programcs > ...
1 namespace exercici14;
2
3 // Declara una paraula i fes que el programa imprimeixi només la primera lletra, l'última lletra i la lletra del mig
4 class Program
5 {
6     0 references
7     static void Main(string[] args)
8     {
9         var paraula = "pastanaga";
10        char primercaracter = paraula[0];
11        char caracterdelmig = paraula[paraula.Length / 2];
12        char caracterfinal = paraula[paraula.Length - 1];
13        Console.WriteLine($"El primer caràcter és {primercaracter}, el caràcter del mig és {caracterdelmig} i el caràcter del final és {caracterfinal}");
14    }
15 }
```

P3 (Execució correcte del codi)

```
isaac@ThinkPad-L590:~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-programes-bàsics-isaacguisset$ dotnet run --project exercicis/exercici14
El primer caràcter és p, el caràcter del mig és a i el caràcter del final és a
isaac@ThinkPad-L590:~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-programes-bàsics-isaacguisset$
```

E15 (Enunciat)

Assigna un nom i un cognom en variables. Imprimeix el nom complet, assegurant-te que la primera lletra de cada paraula estigui en majúscula

P1 (Esquema inicial)

- 1- demanem a l'usuari que introdueixi el seu nom
- 2- llegim el nom introduït per teclat i el guardem en una variable
- 3- demanem a l'usuari que introdueixi el seu cognom
- 4- llegim el cognom introduït i el guardem en una variable
- 5- convertim la primera lletra del nom en majúscula i la resta en minúscules
- 6- convertim la primera lletra del cognom en majúscula i la resta en minúscules
- 7- mostrem per pantalla com eren el nom i el cognom abans de convertir-los
- 8- mostrem per pantalla el nom i el cognom ja convertits amb la primera lletra en majúscula

P2 (Esquema tècnic i codi)

- 1- mostrem un missatge demanant el nom de l'usuari

```
Console.WriteLine("Introdueix el teu nom");
```

- 2- llegim el nom i el guardem en una variable

```
var nom = Console.ReadLine();
```

- 3- mostrem un missatge demanant el cognom

```
Console.WriteLine("Introdueix el teu cognom");
```

- 4- llegim el cognom i el guardem en una variable

```
var cognom = Console.ReadLine();
```

5- transformem el nom perquè tingui la primera lletra en majúscula i la resta en minúscules

```
string nomconvertit = nom.Substring(0, 1).ToUpper() +  
nom.Substring(1).ToLower();
```

6- fem el mateix amb el cognom

```
string cognomconvertit = cognom.Substring(0, 1).ToUpper() +  
cognom.Substring(1).ToLower();
```

7- mostrem com eren el nom i el cognom abans del format

```
Console.WriteLine($"El nom abans del format el nom era {nom}, i el cognom  
era {cognom}");
```

8- mostrem el resultat amb el nom i el cognom correctament formats

```
Console.WriteLine($"El nom convertit és {nomconvertit}, el cognom convertit  
es {cognomconvertit}");
```

codi

```
exercicis > exercici15 > Program.cs > ...  
1 namespace exercici15;  
2  
3 // Assigna un nom i un cognom en variables. Imprimeix el nom complet,  
4 // assegurant-te que la primera lletra de cada paraula estigui en majúscula  
0 references  
5 class Program  
6 {  
    0 references  
7     static void Main(string[] args)  
8     {  
9         Console.WriteLine("Introdueix el teu nom");  
10        var nom = Console.ReadLine();  
11        Console.WriteLine("Introdueix el teu cognom");  
12        var cognom = Console.ReadLine();  
13        string nomconvertit = nom.Substring(0, 1).ToUpper() + nom.Substring(1).ToLower();  
14        string cognomconvertit = cognom.Substring(0, 1).ToUpper() + cognom.Substring(1).ToLower();  
15        Console.WriteLine($"El nom abans del format el nom era {nom}, i el cognom era {cognom}");  
16        Console.WriteLine($"El nom convertit és {nomconvertit}, el cognom convertit es {cognomconvertit}");  
17    }  
18 }  
19 }  
20
```

P3 (Execució correcta del codi)

```
El primer caràcter és p, el caràcter del mig és a i el caràcter del final és a
isaac@ThinkPad-L590:~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-programes-basics-isaacguisset$ dotnet run --project exercicis/exerc
/home/isaac/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-programes-basics-isaacguisset/exercicis/exercici15/Program.cs(13,31): warning
NULL.
/home/isaac/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-programes-basics-isaacguisset/exercicis/exercici15/Program.cs(14,34): warning
NULL.
Introdueix el teu nom
isaac
Introdueix el teu cognom
guisset
El nom abans del format el nom era isaac, i el cognom era guisset
El nom convertit és Isaac, el cognom convertit es Guisset
isaac@ThinkPad-L590:~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-programes-basics-isaacguisset$
```

E16 (Enunciat)

Ningú entén res del nou sistema d'avaluació. Necessitem un programa que calculi la part que és més senzilla, a partir de la mitjana de les notes les pràctiques i la nota de l'exàmen ens calculi la nota final

P1 (Esquema inicial)

- 1- demanem a l'usuari que introdueixi la nota de pràctiques
- 2- llegim la nota de pràctiques i la convertim a número decimal
- 3- demanem a l'usuari que introdueixi la nota de l'examen
- 4- llegim la nota de l'examen i la convertim a número decimal
- 5- calculem la nota final aplicant els percentatges (30% pràctiques i 70% examen)
- 6- arrodonim la nota final al número enter més proper
- 7- mostrem per pantalla la nota final exacta i també la nota arrodonida

P2 (Esquema tècnic i codi)

1- mostrem un missatge demanant la nota de pràctiques

```
Console.Write("Nota de pràctiques: ");
```

2- llegim la nota introduïda i la convertim a double

```
double practiques = double.Parse(Console.ReadLine() ?? "0");
```

3- mostrem un missatge demanant la nota de l'examen

```
Console.Write("Nota de l'examen: ");
```

4- llegim la nota de l'examen i la convertim a double

```
double examen = double.Parse(Console.ReadLine() ?? "0");
```

5- calculem la nota final aplicant els pesos corresponents

```
double notaFinal = practiques * 0.3 + examen * 0.7;
```

6- arrodonim la nota final al valor enter més proper

```
double notaArrodonida = Math.Round(notaFinal);
```

7- mostrem el resultat per pantalla indicant les dues notes notafinal ho poso amb F2 perquè només surtin 2 decimals

```
Console.WriteLine($"La nota final és {notaFinal:F2} o sigui un {notaArrodonida}");
```

```
1 namespace exercici16
2 {
3     0 references
4     class Program
5     {
6         0 references
7         static void Main(string[] args)
8         {
9             Console.Write("Nota de pràctiques: ");
10            double practiques = double.Parse(Console.ReadLine() ?? "0");
11
12            Console.Write("Nota de l'examen: ");
13            double examen = double.Parse(Console.ReadLine() ?? "0");
14
15            double notaFinal = practiques * 0.3 + examen * 0.7;
16
17            double notaArrodonida = Math.Round(notaFinal);
18
19            Console.WriteLine($"La nota final és {notaFinal:F2} o sigui un {notaArrodonida}");
20        }
21    }
```

P3 (Execució correcte del codi)

```
isaac@ThinkPad-L590:~/Escritorio/DAW1/Móduls/Programació/30-9(programes bàsics)/1-programes-basics-isaacguisset$ dotnet run --project exercicis/exercici16
Nota de prácticas: 9,9
Nota de l'examen: 9,9
La nota final és 9,90 o sigui un 10
isaac@ThinkPad-L590:~/Escritorio/DAW1/Móduls/Programació/30-9(programes bàsics)/1-programes-basics-isaacguisset$
```

E17 (Enunciat)

Hi ha una secta d'adoradors dels decimals que volen que els hi feu un programa que a partir d'un nombre real, ex. 4.56, només ens retorni els decimals, 0,56. Als números que només tenen decimals els anomenen "nombres meravellosos"

Número lleig: 23,45
Nombre meravellós: 0,45

Número lleig: 8,5
Nombre meravellós: 0,5

P1 (Esquema inicial)

- 1- Mostrem el missatge a l'usuari per demanar que introdueixi un número.
- 2- Llegim el número introduït i el convertim a double amb double.Parse.
- 3- Calculem només la part decimal restant la part entera (Math.Floor(nombrelleig)) del número original.
- 4- Arrodonim el resultat a 2 decimals amb Math.Round.
- 5- Mostrem per pantalla el número lleig i el nombre meravellós.

P2 (Esquema tècnic i codi)

- 1- `Console.Write("Introdueix un número: ");`
Mostra el missatge per demanar el número.
- 2- `double nombrelleig = double.Parse(Console.ReadLine());`
Llegeix l'entrada i la converteix a double.
- 3- `double nombremaravellós = nombrelleig - Math.Floor(nombrelleig);`
Calcula la part decimal restant la part entera.
- 4- `nombremaravellós = Math.Round(nombremaravellós, 2);`
Arrodoneix la part decimal a 2 decimals.
- 5- `Console.WriteLine($"El número lleig és {nombrelleig}");`
Mostra el número original.
- 6- `Console.WriteLine($"Nombre meravellós: {nombremaravellós:F2}");`
Mostra el número meravellós amb 2 decimals.

codi

```
exercicis > exercici17 > Program.cs > ...
9  Número lleig: 8,5
10 Nombre meravellós: 0,5
11 */
12
13 namespace exercici17
14 {
15     0 references
16     class Program
17     {
18         0 references
19         static void Main(string[] args)
20         {
21             Console.Write("Introdueix un número: ");
22             double nombrelleig = double.Parse(Console.ReadLine());
23
24             double nombremaravellós = nombrelleig - Math.Floor(nombrelleig);
25
26             nombremaravellós = Math.Round(nombremaravellós, 2);
27
28             Console.WriteLine($"El número lleig és {nombrelleig}");
29             Console.WriteLine($"Nombre meravellós: {nombremaravellós:F2}");
30         }
31     }
32 }
```


P3 (Execució correcte del codi)

```
isaac@ThinkPad-L590:~/Escritorio/DAW1/Móduls/Programació/30-9(programes bàsics)/1-progr  
/home/isaac/Escritorio/DAW1/Móduls/Programació/30-9(programes bàsics)/1-programes-basic  
parámetro "s" en "double double.Parse(string s)".  
Introduceix un número: 12,88  
El número lleig és 12,88  
Nombre meravellós: 0,88  
isaac@ThinkPad-L590:~/Escritorio/DAW1/Móduls/Programació/30-9(programes bàsics)/1-progr
```

E18 (Enunciat)

Un rellotger està perdent la memòria i necessita un programa que a partir de l'hora actual li puguem dir quina hora serà d'aquí un nombre determinat d'hores.

Hora actual: 9

Hores a incrementar: 3

D'aquí a 3 hores seran les 12

Hora actual: 11

Hores a incrementar: 3

D'aquí a 3 hores seran les 2

P1 (Esquema inicial)

- 1- Mostrem un missatge per demanar a l'usuari l'hora actual.
- 2- Llegim el valor introduït i el guardem a la variable `hora_actual`.
- 3- Mostrem un missatge per demanar quantes hores volem incrementar.
- 4- Llegim aquest valor i el guardem a la variable `hores_a_incrementar`.
- 5- Calculem l'hora final sumant les dues variables i aplicant el mòdul 24 per mantenir el format horari.
- 6- Mostrem per pantalla quina hora serà després de sumar les hores.

P2 (Esquema tècnic i codi)

- 1- `Console.Write("Hora actual: ");` → Mostra el missatge per demanar l'hora actual.
- 2- `int hora_actual = Convert.ToInt16(Console.ReadLine());` → Llegeix l'hora introduïda per l'usuari.
- 3- `Console.Write("Hores a incrementar: ");` → Mostra el missatge per demanar les hores a sumar.
- 4- `int hores_a_incrementar = Convert.ToInt16(Console.ReadLine());` → Llegeix el nombre d'hores a incrementar.
- 5- `int hora_final = (hora_actual + hores_a_incrementar) % 24;` → Suma les hores i aplica el mòdul 24 per ajustar l'hora dins del dia.

6- `Console.WriteLine($"D'aquí a {hores_a_incrementar} hores seran les {hora_final}");` → Mostra el resultat final per pantalla.

```
5
6
7 {
8     0 references
9     class Program
10    {
11        0 references
12        static void Main(string[] args)
13        {
14            Console.Write("Hora actual: ");
15            int hora_actual = Convert.ToInt16(Console.ReadLine());
16
17            Console.Write("Hores a incrementar: ");
18            int hores_a_incrementar = Convert.ToInt16(Console.ReadLine());
19
20            // utilitzem el modul
21            int hora_final = (hora_actual + hores_a_incrementar) % 24;
22
23            Console.WriteLine($"D'aquí a {hores_a_incrementar} hores seran les {hora_final}");
24        }
25    }
26 }
```

P3 (Execució correcte del codi)

```
isaac@ThinkPad-L590:~/Escritorio/DAW1/Móduls/Programació/30-9(programes bàsics)/1-progr
Hora actual: 17
Hores a incrementar: 8
D'aquí a 8 hores seran les 1
isaac@ThinkPad-L590:~/Escritorio/DAW1/Móduls/Programació/30-9(programes bàsics)/1-progr
```

E19 (Enunciat)

Assigna una paraula a una variable de text. Utilitza una funció per a obtenir la seva longitud i imprimeix la paraula i el seu nombre de caràcters.

P1 (Esquema inicial)

- 1- Determinar el valor de la variable paraula
- 2- Determinar el valor de la variable caracters en funcio de la paraula
- 3- Retornar per pantalla els caràcters de la paraula

P2 (Esquema tècnic i codi)

- 1- Definir la variable amb `string paraula = "supercalifragilisticospialidoso;"`
- 2- Definir la variable caracters amb `int caracters = paraula.Length;`
- 3- Fer un print de text+variables `Console.WriteLine($"La paraula {paraula} té {caracters} caracters");`

```
1 namespace exercici19;
2
3 // Assigna una paraula a una variable de text. Utilitza una funció per a obtenir la seva longitud i
4 // imprimeix la paraula i el seu nombre de caràcters.
5 class Program
6 {
7     static void Main(string[] args)
8     {
9         string paraula = "supercalifragilisticospialidoso";
10        int caracters = paraula.Length;
11        Console.WriteLine($"La paraula {paraula} té {caracters} caracters");
12    }
13 }
```

P3 (Execució correcte del codi)

```
● guisset$ dotnet run --project exercicis/exercici19
La paraula supercalifragilisticospialidoso té 31 caracters
isaac@ThinkPad-L590:~/Escritorio/DAW1/Móduls/Programació/30-9(p
● guisset$
```

E20 (Enunciat)

Ens cal un programa que quan li entrem un número de tres xifres i ens retorni el número amb els dígitos invertits.

Ex. 245 -> 542

Entra el número a invertir: 347

El número invertit és: 743

P1 (Esquema inicial)

1-Demanar el numero a invertir i recollir-lo

2-convertir el número a una cadena de caràcters

3-invertir l'ordre

4- retornar per pantalla l'ordre invertit

P2 (Esquema tècnic i codi)

1- definir una variable de text per guardar el número que escriu l'usuari

`string numero = Console.ReadLine();`

2-Convertir el número en un array de caràcters `char[] digits = numero.ToCharArray();`

3- Invertir aquest array: `Array.Reverse(digits);`

4- Tornar a crear una cadena amb els caràcters invertits: `string numeroInvertit = new string(digits);`

codi

```
12  class Program
13  {
14      0 references
15      static void Main(string[] args)
16      {
17          Console.Write("Entra el número a invertir: ");
18          string numero = Console.ReadLine();
19
20          char[] digits = numero.ToCharArray();
21          Array.Reverse(digits);
22
23          string numeroInvertit = new string(digits);
24
25          Console.WriteLine($"El número invertit és: {numeroInvertit}")
26      }
27  }
```

P3 (Execució correcte del codi)

```
isaac@ThinkPad-L590:~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-  
programes-basics-isaacguisset$ dotnet run --project exercicis/exercici20  
Entra el número a invertir: 12  
El número invertit és: 21  
isaac@ThinkPad-L590:~/Escritorio/DAW1/Mòduls/Programació/30-9(programes bàsics)/1-  
programes-basics-isaacguisset$
```

Resum dels aprenentatges i conclusions

Al llarg d'aquests 20 exercicis d'iniciació a la programació en C#, he pogut aprendre i posar en pràctica diversos conceptes inicials molt interessants sobre la programació en general

Declaració i ús de variables: he après a definir variables, assignar-los valors i entendre els diferents tipus de dades (string, int, double, char, bool...).

Entrada i sortida d'informació: he practicat la manera de capturar dades introduïdes per l'usuari mitjançant `Console.ReadLine()` i com mostrar resultats amb `Console.WriteLine()`.

Constants i conversions de tipus: he entès la diferència entre variables modificables i constants (`const`) i com canviar de tipus de dades en temps d'execució (`Convert`, `Parse`).

Operacions matemàtiques i lògiques: he aplicat sumes, restes, divisions, mòduls i arrodoniments (`Math.Round`, `%`) així com condicions amb `if-else` i operadors booleans.

Gestió de text i caràcters: he practicat amb cadenes (`string.Length`, `Substring`, concatenació, interpolació de cadenes, majúscules i minúscules) i amb arrays de caràcters (`ToCharArray`, `Array.Reverse`).

Ús de booleans i operadors ternaris: he entès com simplificar condicions i representar estats amb `true/false`.

Estructures de control: encara que no les domino del tot, he començat a treballar amb condicions i m'he adonat de la importància dels bucles per optimitzar programes repetitius.

Validació i tractament d'errors: he vist com prevenir errors de format amb `TryParse` o amb blocs `try-catch`.