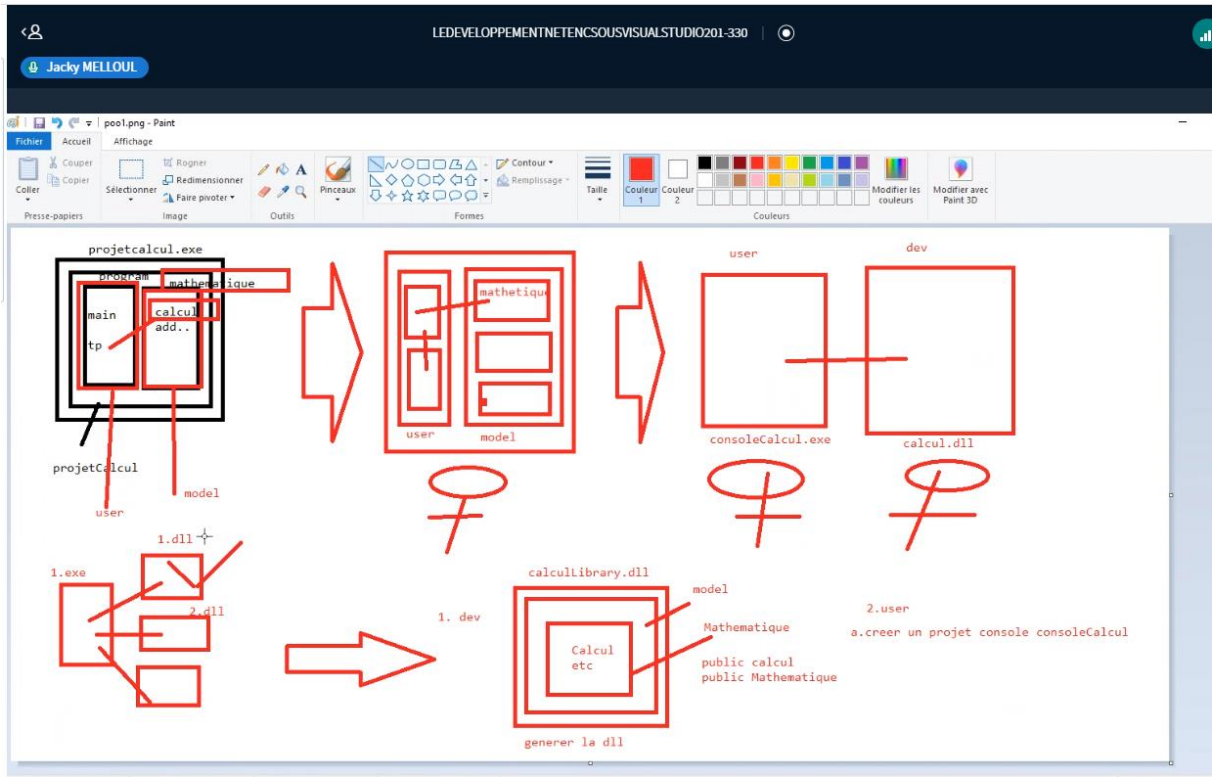
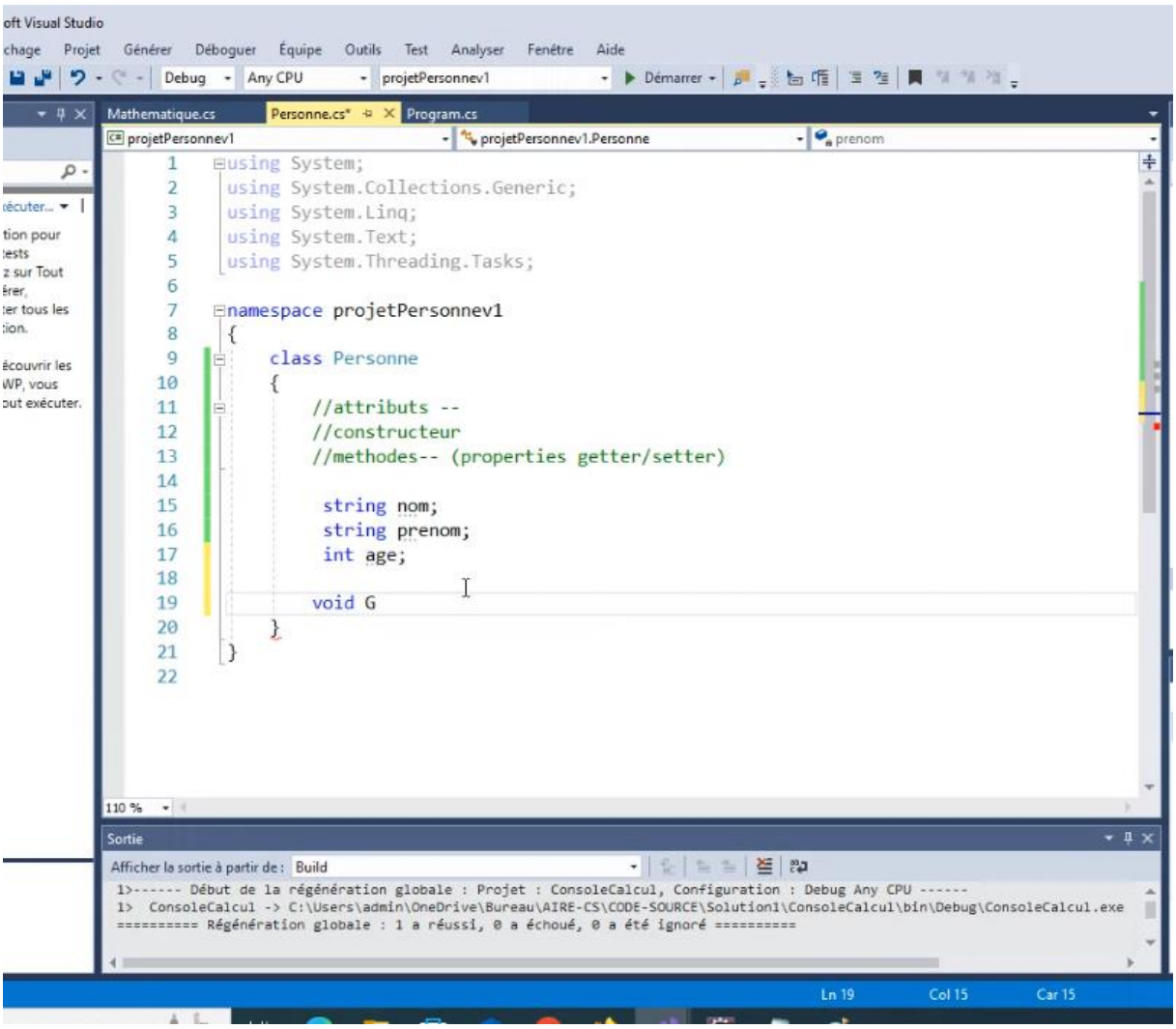
Programation OO



4762

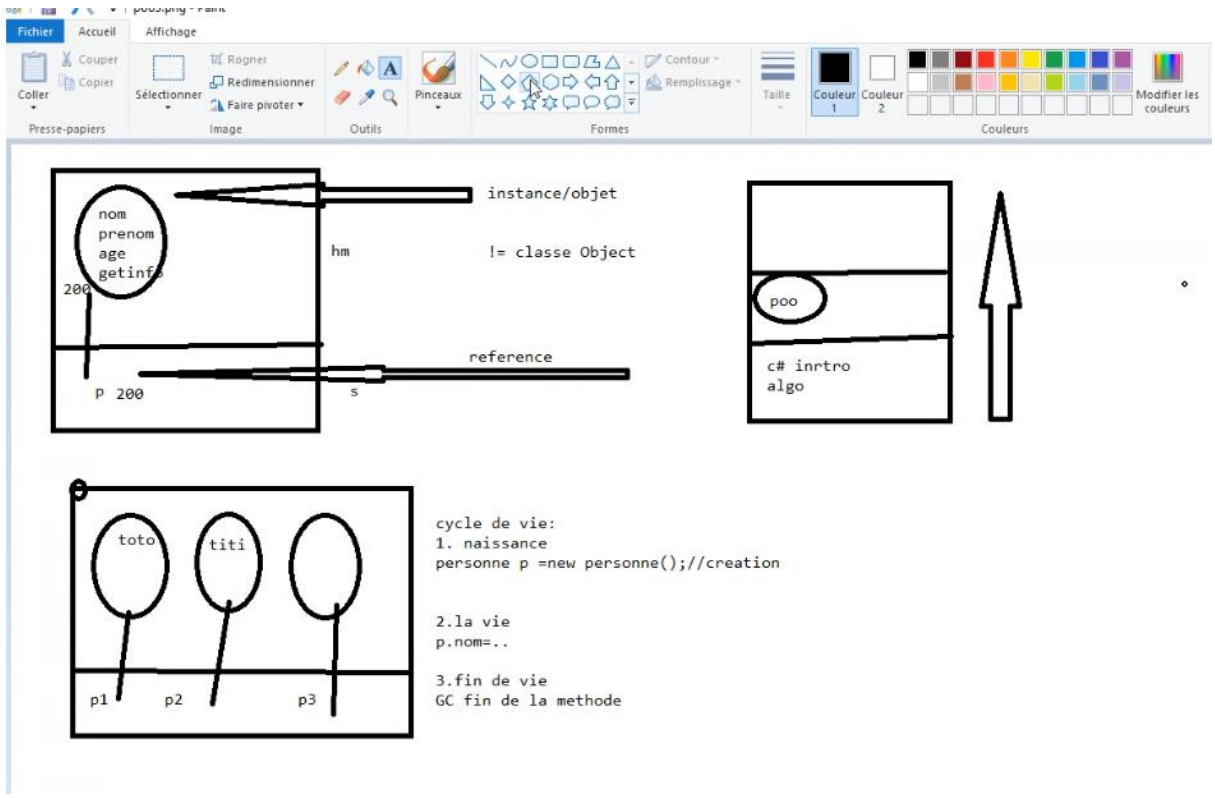

oft Visual Studio
chage Projet Générer Déboguer Équipe Outils Test Analyser Fenêtre Aide
Debug
Any CPU
projetPersonnev1
Démarrer
Mathematique.cs
Personne.cs*
Program.cs
projetPersonnev1
projetPersonnev1.Personne
prenom
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace projetPersonnev1
8 {
9 class Personne
10 {
11 //attributs --
12 //constructeur
13 //methodes-- (properties getter/setter)
14
15 string nom;
16 string prenom;
17 int age;
18
19 void G
20 }
21 }
22
110 %
Sortie
Afficher la sortie à partir de : Build
1>------ Début de la régénération globale : Projet : ConsoleCalcul, Configuration : Debug Any CPU ------
1> ConsoleCalcul -> C:\Users\admin\OneDrive\Bureau\AIRE-CS\CODE-SOURCE\Solution1\ConsoleCalcul\bin\Debug\ConsoleCalcul.exe
========== Régénération globale : 1 a réussi, 0 a échoué, 0 a été ignoré ==========
Ln 19
Col 15
Car 15

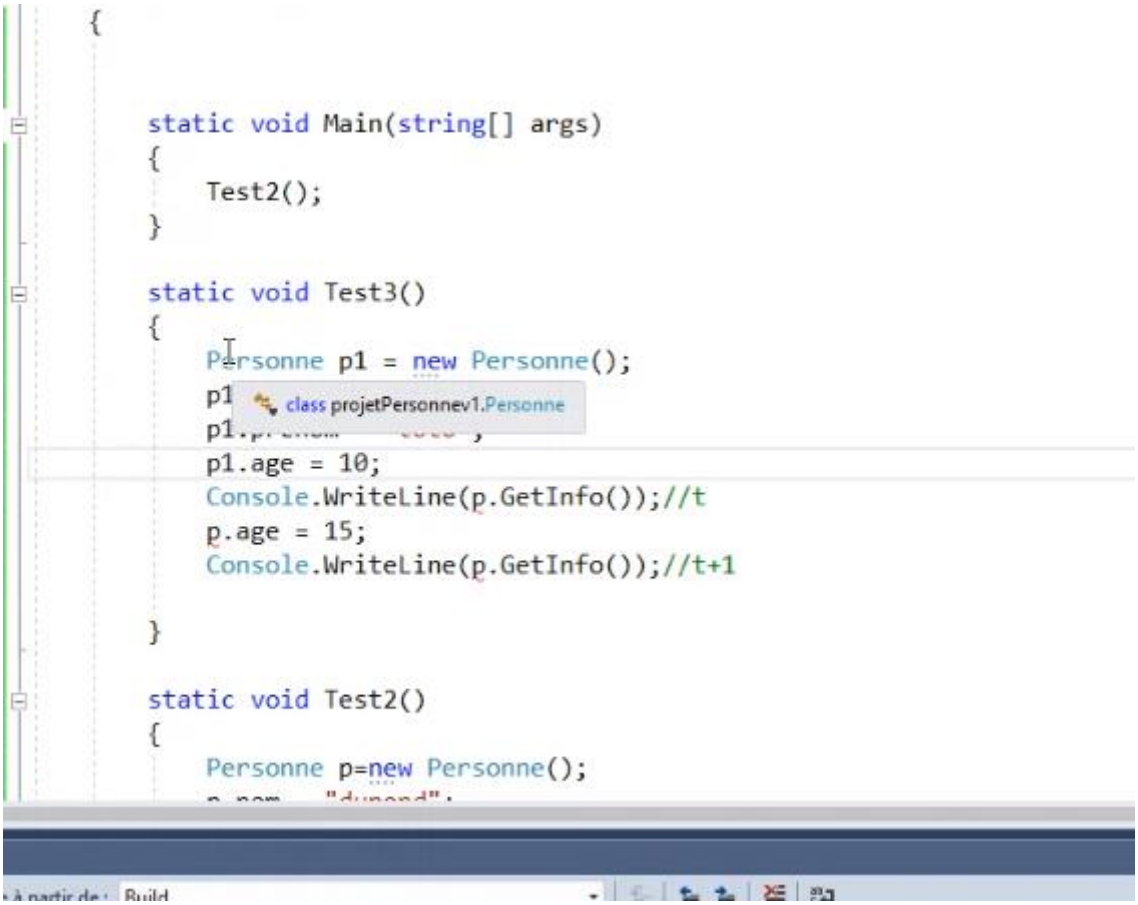
```
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace projetPersonnev1
{
    class Program
    {
        static void Main(string[] args)
        {
            Test1();
        }
        static void Test1()
        {
            int a;
            string str; // declaration
            str = "toto";// initialisation

            Personne p; // declaration
            p = new Personne();//instance ++
            // initialisation
```

```
namespace projetPersonnev1
{
    class Personne
    {

        //attributs --variable global --variable de la classe
        public string nom;//null != ""
        public string prenom;
        public int age;//0


        //constructeur:
        //1. implicite non codé
        //2. constructeur par defaut: sans parametre

        public Personne()
        {
            Console.WriteLine("coucou");
        }


        //methodes-- (properties getter/setter)
        public string  GetInfo()
        {
            string reponse = "";
            reponse += (nom == null) ? " NOM NULL" : nom+"  ";
```

```
Afficher la sortie à partir de : Build
projetPersonnev1 -> C:\Users\admin\OneDrive\Bureau\AIRE-CS\CODE-SOURCE\Solution1\projetPersonnev1\bin\Debug\projetPersonn
```

```
amespace projetPersonnev1

    class Program
    {


        static void Main(string[] args)
        {
            Test2();
        }

        static void Test2()
        {
        }
            static void Test1()
        {
            int a;
            string str; // declaration
            str = "toto";// initialisation

            Personne p; // declaration
            p = new Personne();//instance ++ appel au constructeur
            // initialisation
            Console.WriteLine(p.GetInfo());
```



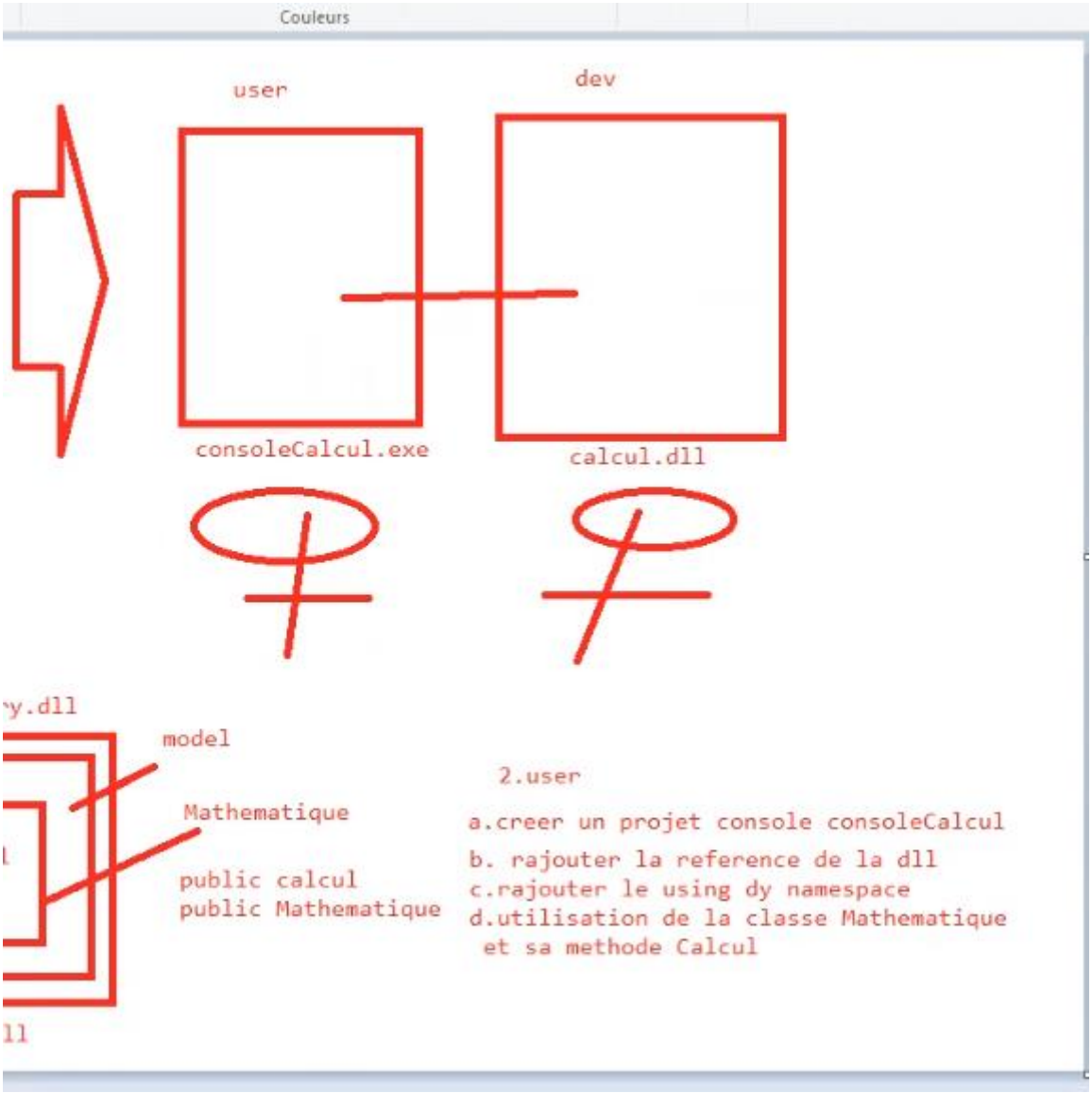


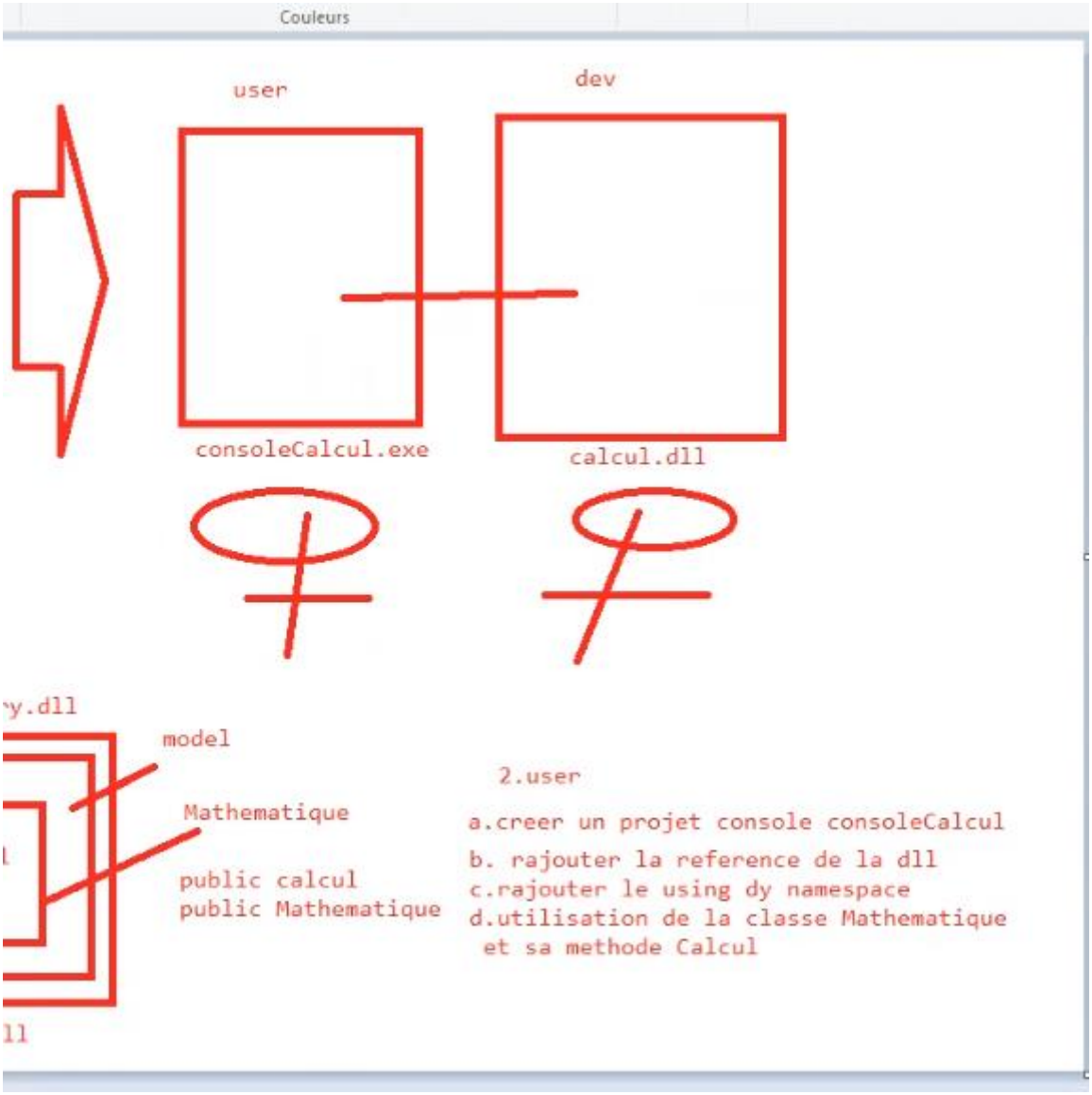

```
using System;

namespace tpVille
{
class Program
{
static void Main(string[] args)
{
Test1();
Test2();
Test3();
Console.ReadKey();
}

// Test avec une ville
static void Test1()
{
Ville paris = new Ville("Paris", "France", 2.1);
Console.WriteLine(paris.GetInfos());
}

// Test avec deux villes en utilisant les constructeurs
static void Test2()
{
Ville newYork = new Ville("New York", "USA", 8.3);
```

```
Ville tokyo = new Ville("Tokyo", "Japon", 9.2);
Console.WriteLine(newYork.GetInfos());
Console.WriteLine(tokyo.GetInfos());
}

// Test avec un tableau de villes
static void Test3()
{
Ville[] villes = new Ville[]
{
new Ville("São Paulo", "Brésil", 12.2),
new Ville("Londres", "Royaume-Uni", 8.9),
new Ville("Berlin", "Allemagne", 3.6)
};

foreach (var ville in villes)
{
Console.WriteLine(ville.GetInfos());
}
}
}
}
```

```
using System;

namespace tpVille
{
public class Ville
{
public string nom;
public string pays;
public double nbHabitant; // en millions

// Constructeur par défaut
public Ville()
{
}

// Constructeur d'initialisation avec tous les paramètres
public Ville(string nom, string pays, double nbHabitant)
{
this.nom = nom;
this.pays = pays;
this.nbHabitant = nbHabitant;
}

// Méthode non publique pour déterminer la taille de la ville
private string TailleVille()
```

```
{
if (nbHabitant < 5)
return "petite";
else if (nbHabitant >= 5 && nbHabitant < 10)
return "moyenne";
else
return "grande";
}

// Méthode pour obtenir les informations de la ville
public string GetInfos()
{
return nom.ToLower() + " " + pays.ToUpper() + " " + nbHabitant + "M - " + TailleVille();
}
}
}
```


Couleurs
user
dev
consoleCalcul.exe
calcul.dll
y.dll
model
Mathematique
public calcul
public Mathematique
ll
2.user
a.creer un projet console consoleCalcul
b. rajouter la reference de la dll
c.rajouter le using dy namespace
d.utilisation de la classe Mathematique
et sa methode Calcul



les constructeurs sont toujour privée ; pour les methode ca depend, les attributs en privé

value = valeur de l'exterieur

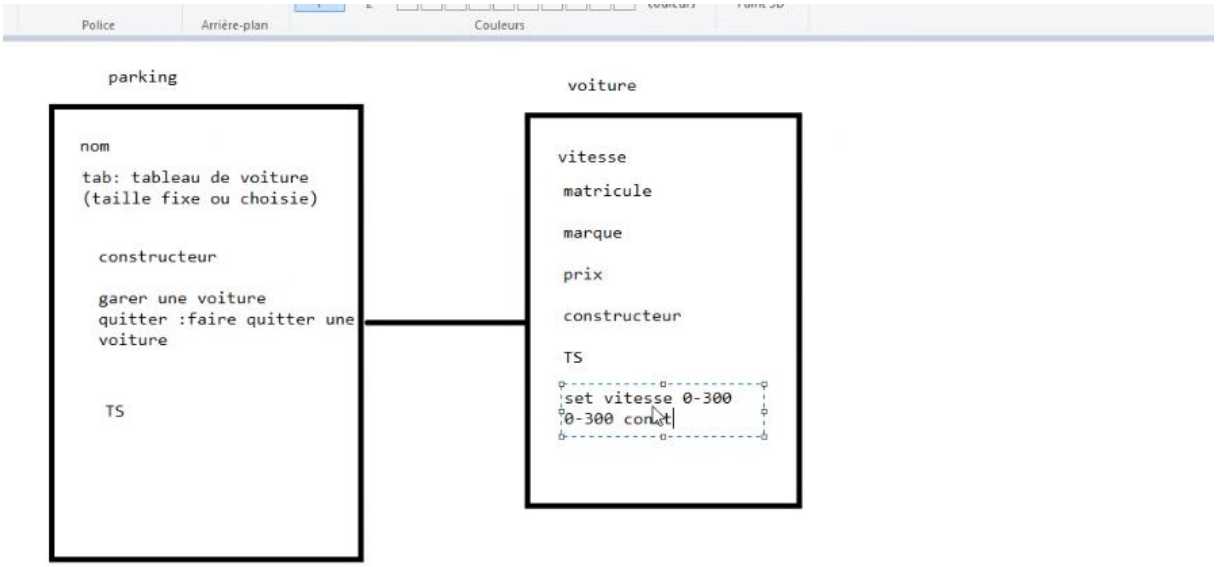
personnev3

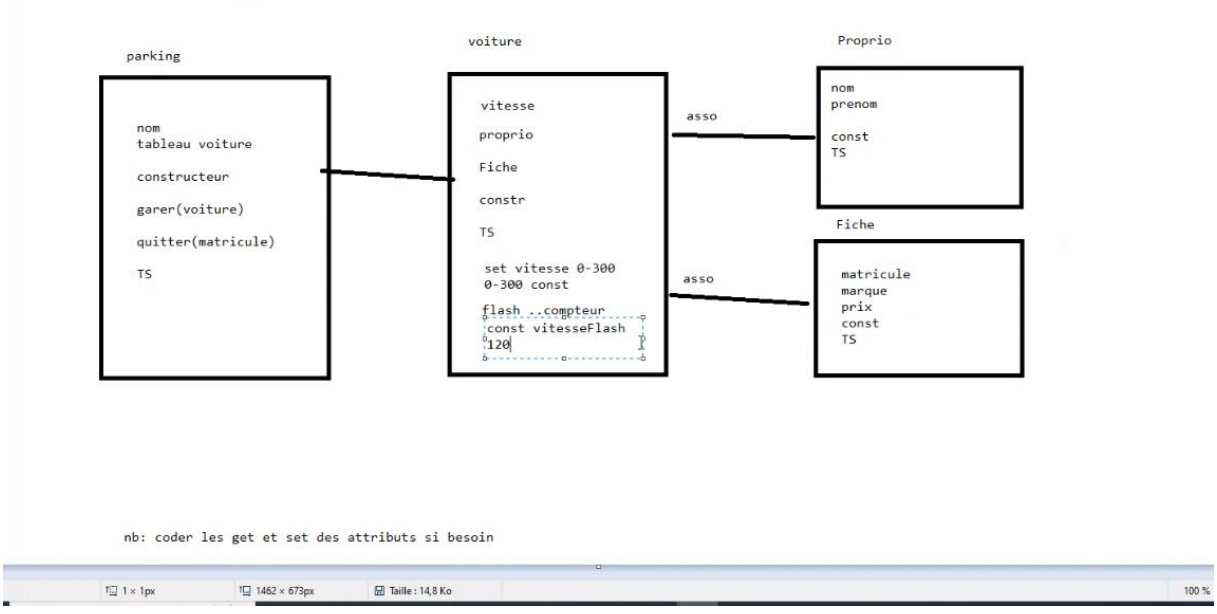
```
private string nom;
        private string prenom;
        private int age;

        public Personne(string nom, string prenom, int age)
        {
            this.nom = nom;
```

```
            this.prenom = prenom;
            this.Age = age;

        }
        public Personne()
        {

        }

        //property Age get et set
        public int Age
        {

            //lecture
            get
            {
                return age;
            }
            //ecriture 0 et 120
            set
            {
                if (value > 0 && value < 120)
                    age = value;
            }
        }

        public override string ToString()
        {
            return this.nom + "   " + prenom + "  " + age;
        }
```

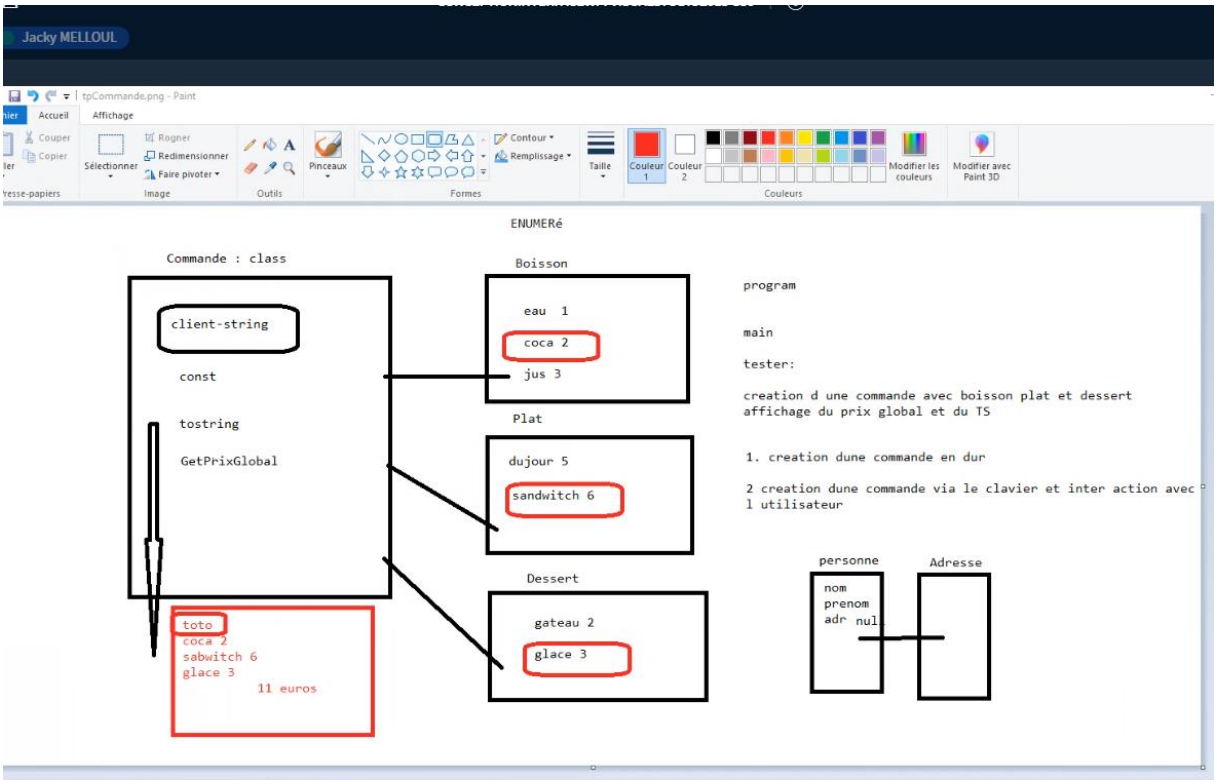
Je n'ai pas utilisé de get et set car il n'y a pas de condition sur la lecture et modification

```
public bool Supprimer(int id)
        {

            bool reponse = false;
            for (int i = 0; i < tab.Length; i++)
                if (tab[i] != null && id == tab[i].Id)
                {
                    tab[i] = null;
                    reponse = true;
                }
            return reponse;

        }
```





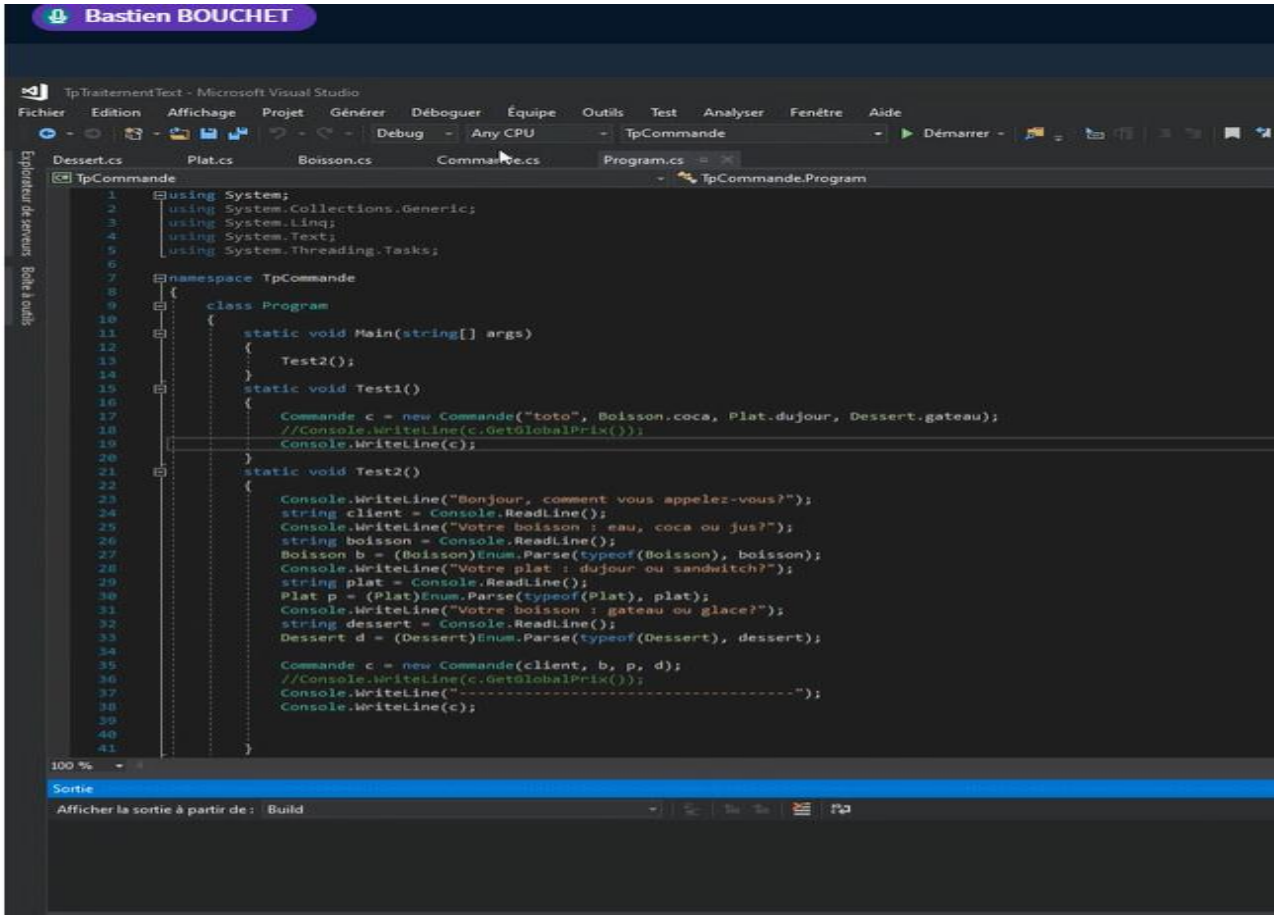
nb: coder les get et set des attributs si besoin

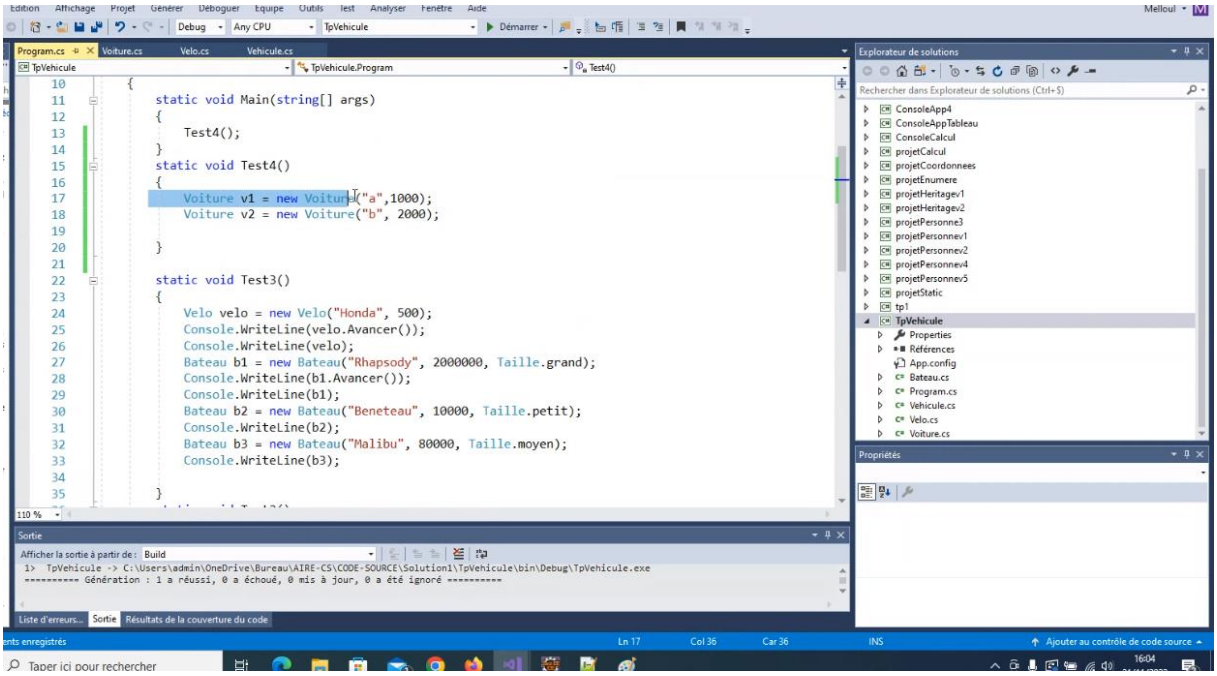

Jacky MELLOUL
tpCommande.png - Paint
ENUMERé
Commande : class
client-string
const
tostring
GetPrixGlobal
toto
coca 2
sabwitch 6
glace 3
11 euros
Boisson
eau 1
coca 2
jus 3
Plat
dujour 5
sandwitch 6
Dessert
gateau 2
glace 3
program
main
tester:
creation d une commande avec boisson plat et dessert
affichage du prix global et du TS
1. creation dune commande en dur
2 creation dune commande via le clavier et inter action avec
l utilisateur
personne
nom
prenom
adr nul
Adresse

5-

)

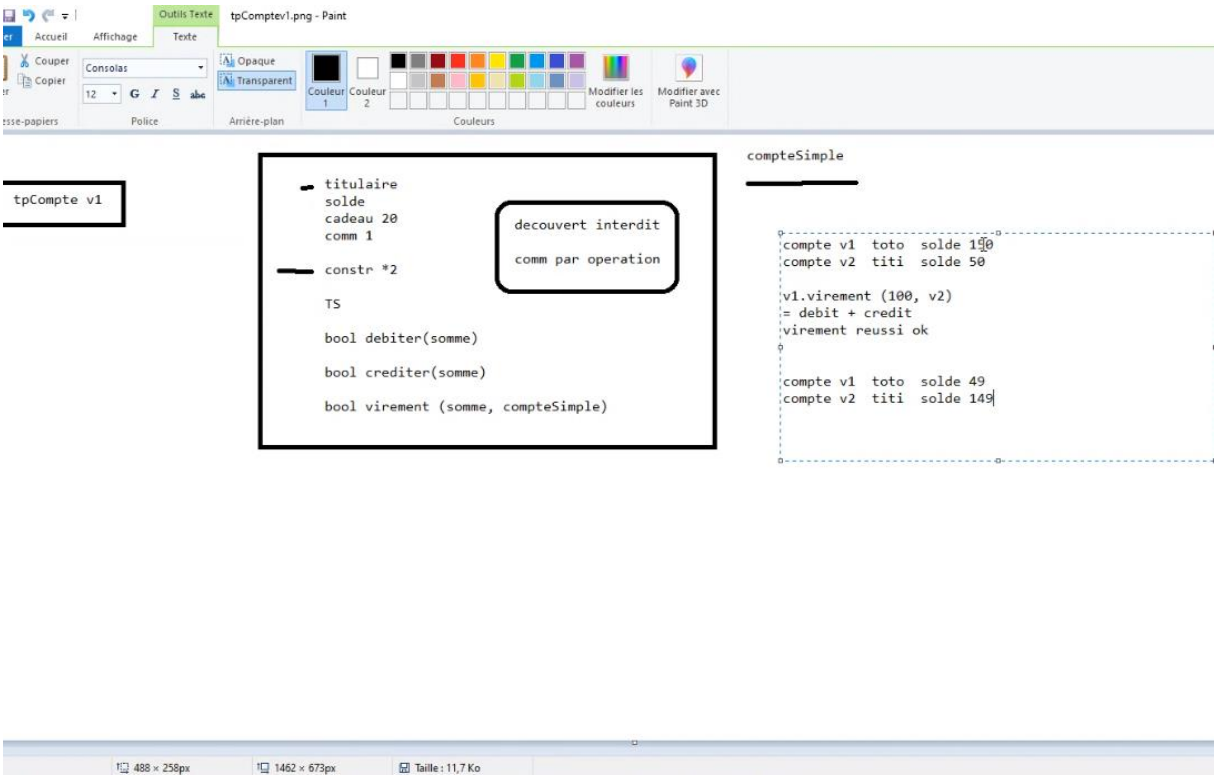
Use

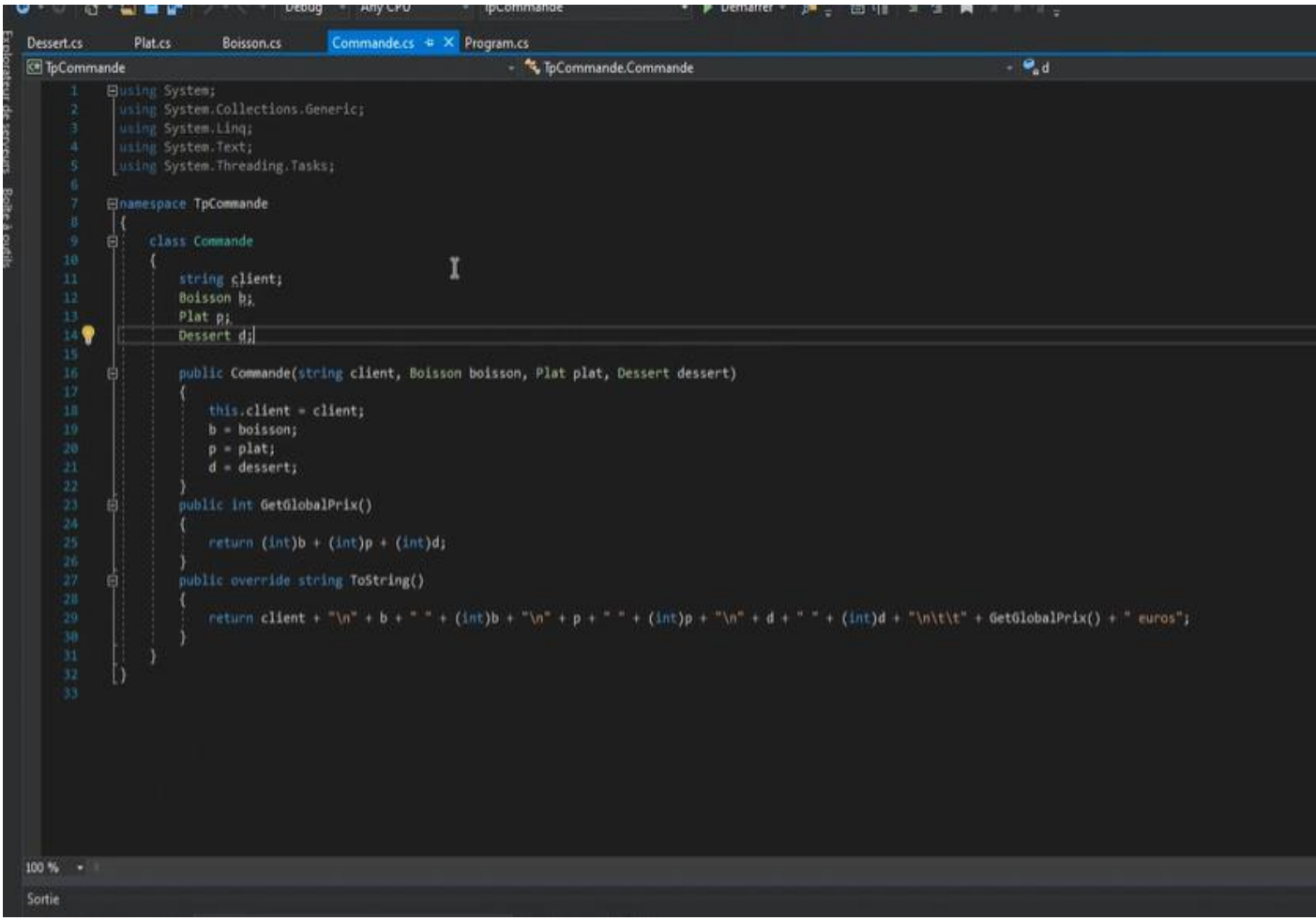

Bastien BOUCHET
TpTraitementText - Microsoft Visual Studio
Fichier Edition Affichage Projet Générer Déboguer Équipe Outils Test Analyser Fenêtre Aide
Debug Any CPU TpCommande Démarrer
Dessert.cs Plat.cs Boisson.cs Commande.cs Program.cs
TpCommande TpCommande.Program
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace TpCommande
{
class Program
{
static void Main(string[] args)
{
Test2();
}
static void Test1()
{
Commande c = new Commande("toto", Boisson.coca, Plat.dujour, Dessert.gateau);
//Console.WriteLine(c.GetGlobalPrix());
Console.WriteLine(c);
}
static void Test2()
{
Console.WriteLine("Bonjour, comment vous appelez-vous?");
string client = Console.ReadLine();
Console.WriteLine("Votre boisson : eau, coca ou jus?");
string boisson = Console.ReadLine();
Boisson b = (Boisson)Enum.Parse(typeof(Boisson), boisson);
Console.WriteLine("Votre plat : dujour ou sandwitch?");
string plat = Console.ReadLine();
Plat p = (Plat)Enum.Parse(typeof(Plat), plat);
Console.WriteLine("Votre boisson : gateau ou glace?");
string dessert = Console.ReadLine();
Dessert d = (Dessert)Enum.Parse(typeof(Dessert), dessert);
Commande c = new Commande(client, b, p, d);
//Console.WriteLine(c.GetGlobalPrix());
Console.WriteLine("-----------------------------------");
Console.WriteLine(c);
}
}
100 %
Sortie
Afficher la sortie à partir de : Build



\n\t\t

```
static void Test4()
      {
         Voiture v1=new Voiture("a",1000);
         v1.GpsONOff();

         Vehicule v2 = new Voiture("b", 2000);


        //  Voiture v3 = new  Vehicule("b", 2000);

      }
```


tpComptev1.png - Paint
Outils Texte
Accueil
Affichage
Texte
Couper
Copier
Consolas
12
Police
Opaque
Transparent
Arrière-plan
Couleur 1
Couleur 2
Couleurs
Modifier les couleurs
Modifier avec Paint 3D
tpCompte v1
titulaire
solde
cadeau 20
comm 1
constr *2
TS
bool debiter(somme)
bool crediter(somme)
bool virement (somme, compteSimple)
decouvert interdit
comm par operation
compteSimple
compte v1 toto solde 100
compte v2 titi solde 50
v1.virement (100, v2)
= debit + credit
virement reussi ok
compte v1 toto solde 49
compte v2 titi solde 149
488 × 258px
1462 × 673px
Taille : 11,7 Ko


Dessert.cs
Plat.cs
Boisson.cs
Commande.cs
Program.cs
TpCommande
TpCommande.Commande
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace TpCommande
{
class Commande
{
string client;
Boisson b;
Plat p;
Dessert d;
public Commande(string client, Boisson boisson, Plat plat, Dessert dessert)
{
this.client = client;
b = boisson;
p = plat;
d = dessert;
}
public int GetGlobalPrix()
{
return (int)b + (int)p + (int)d;
}
public override string ToString()
{
return client + "\n" + b + " " + (int)b + "\n" + p + " " + (int)p + "\n" + d + " " + (int)d + "\n\t\t" + GetGlobalPrix() + " euros";
}
}
}
100 %
Sortie