Exception

class Program

{

static void Main(string[] args)

{

try

{

string chaine = "dix";

int valeur = Convert.ToInt32(chaine);

}

catch (Exception)

{

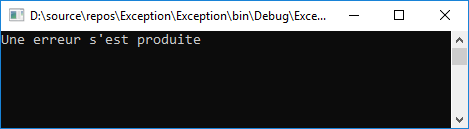
Console.WriteLine("Une erreur s'est produite");

}

Console.Read();

}

}



# Quelques méthodes de l’objet Exception

class Program

{

static void Main(string[] args)

{

try

{

string chaine = "dix";

int valeur = Convert.ToInt32(chaine);

}

catch (Exception ex)

{

Console.WriteLine("Il y a eu une erreur, plus d'information ci-dessous :");

Console.WriteLine("\nMessage d'erreur : "+ex.Message);

Console.WriteLine("\nPile d'appel : "+ex.StackTrace);

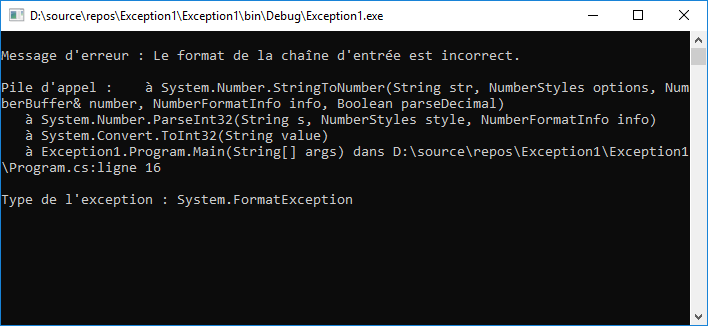
Console.WriteLine("\nType de l'exception : " + ex.GetType());

}

Console.Read();

}

}



class Program

{

static void Main(string[] args)

{

try

{

string chaine = "dix";

int valeur = Convert.ToInt32(chaine);

}

catch (Exception ex)

{

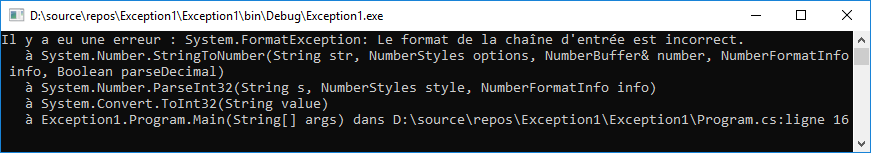
Console.WriteLine("Il y a eu une erreur : "+ ex.ToString());

}

Console.Read();

}

}



class Program

{

static void Main(string[] args)

{

try

{

string chaine = "dix";

int valeur = Convert.ToInt32(chaine);

}

catch (FormatException ex)

{

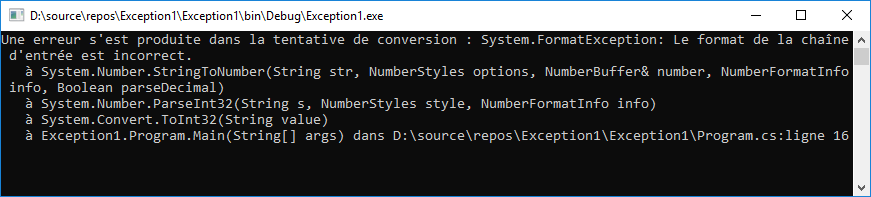
Console.WriteLine("Une erreur s'est produite dans la tentative de conversion : "+ ex);

}

Console.Read();

}

}



class Program

{

public static double RacineCarree(double valeur)

{

if (valeur <= 0)

throw new ArgumentOutOfRangeException("valeur", "le paramètre doit être positif");

return Math.Sqrt(valeur);

}

static void Main(string[] args)

{

Console.WriteLine("Saisie de la chaîne :");

string chaine = Console.ReadLine();

try

{

if (chaine == "null")

{

var car = new Automobile();

Console.WriteLine(car.Engine.ToString());

}

else

{

double valeur = Convert.ToDouble(chaine);

Console.WriteLine(RacineCarree(valeur));

Console.WriteLine("Ok");

}

}

catch (FormatException ex)

{

Console.WriteLine("Erreur de format : "+ ex);

}

catch (NullReferenceException ex)

{

Console.WriteLine("Erreur de référence nulle : " + ex);

}

catch (SystemException ex)

{

Console.WriteLine("Erreur système autres que FormatException et NullReferenceException : " + ex);

}

catch (Exception ex)

{

Console.WriteLine("Toutes les autres exceptions : " + ex);

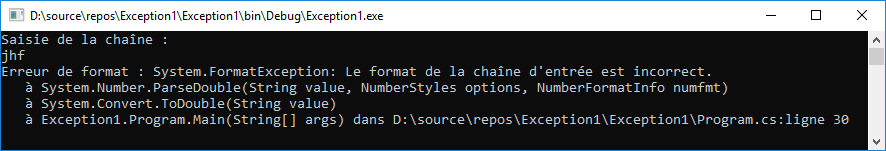
}

Console.Read();

}

}

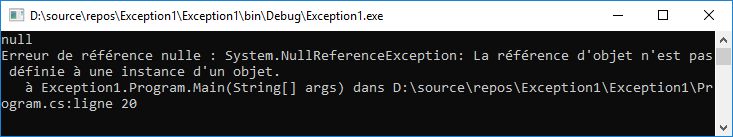
# FormatException



# NullReferenceException

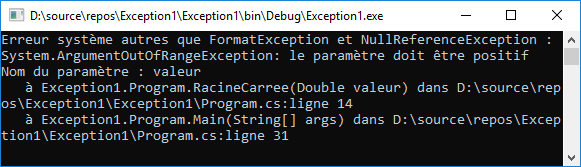
Quand chaine = Console.ReadLine() vaut "null", on entre dans la condition qui crée un objet Automobile vide.

class Automobile { public EngineInfo Engine { get; set; } }



# Lever une exception avec throw

Lorsque la valeur est inférieur ou égal à 0 dans la fonction RacineCarre, on renvoi un throw qui est intercepté dans catch SystemException.



# Exception connexion à une base de données et mot clé finally

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using MySql.Data.MySqlClient;

namespace ExceptionFinally

{

class Program

{

static void Main(string[] args)

{

MySqlConnection co;

string connect = "Server=127.0.0.1;Database=sakila;Uid=root;Pwd=;";

co = new MySqlConnection(connect);

try

{

// Ouverture de la connexion à la base de données

co.Open();

// Exécution d’une instruction en base de données

string query = "INSERT INTO actor(first\_name, last\_name) VALUES ('joe', 'dump');";

MySqlCommand cmd = new MySqlCommand(query, co);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

Console.WriteLine("Connexion impossible, raisons : " + ex);

}

finally

{

// Fermeture de la base de données dans tout les cas

try { co.Close(); }

catch (MySqlException e) { Console.WriteLine(e); }

}

Console.Read();

}

}

}

# MailException et Fichier log

class Program

{

static void Main(string[] args)

{

try

{

string chaine = "dix";

int valeur = Convert.ToInt32(chaine);

}

catch (Exception ex)

{

Console.WriteLine("L'application a rencontré un problème, un mail a été envoyé à l'administrateur...");

Console.WriteLine(ex);

LogHelper.WriteToFile(ex.Message, "Program");

MailException mail = new MailException();

string erreur = Convert.ToString(ex);

mail.sendMail("mat.planchot@gmail.com", erreur);

}

Console.Read();

}

}

class LogHelper

{

public static void WriteToFile(string errorMessage, string className)

{

DateTime nowDate = DateTime.Now;

string shortDate = String.Format("{0:yyyy-MM-dd}", nowDate);

string filename = string.Format("{0}.log", shortDate);

//création du nom de fichier .log

filename = filename.Replace("/", "-");

//récupérer le path complet, application PC

string rootPath = Path.GetFullPath("E:/source/repos/CoursException/");

//ou le path pour une application serveur

//System.Web.HttpContext.Current.Server.MapPath("~/Data/Log/");

string fullFilename = string.Format(@"{0}{1}", rootPath, filename);

//vérifier les dossiers Data & Log

if (!Directory.Exists(rootPath))

{

//création du dossier

Directory.CreateDirectory(rootPath);

}

//vérifier le fichier

if (!System.IO.File.Exists(fullFilename))

{

//création du fichier log du jour

System.IO.FileStream f = System.IO.File.Create(fullFilename);

f.Close();

}

using (StreamWriter writer = new StreamWriter(fullFilename, true))

{

//écriture dans le fichier log du jour

writer.WriteLine(string.Format(

"[{0} ON {1}] : {2}",

DateTime.Now,

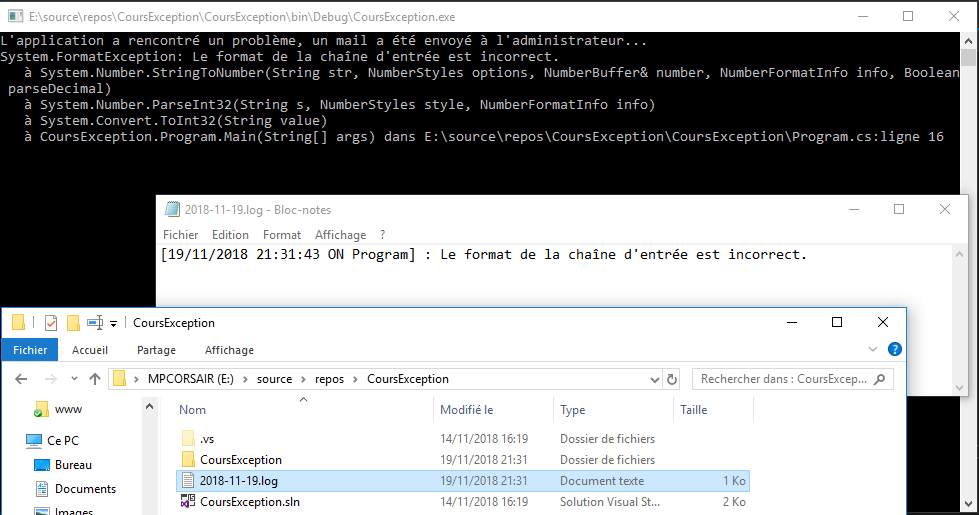
className,

errorMessage));

}

}

}



class MailException

{

public void sendMail(string to, string message)

{

SmtpClient client = new SmtpClient();

client.Port = 587;

client.Host = "SSL0.OVH.NET";

client.EnableSsl = true;

client.Timeout = 10000;

client.DeliveryMethod = SmtpDeliveryMethod.Network;

client.UseDefaultCredentials = false;

client.Credentials = new System.Net.NetworkCredential("matthieu.planchot@stjosup.com", "pm18041994");

MailMessage mm = new MailMessage("matthieu.planchot@stjosup.com", to, "Une erreur s'est produite", message);

mm.BodyEncoding = UTF8Encoding.UTF8;

mm.DeliveryNotificationOptions = DeliveryNotificationOptions.OnFailure;

client.Send(mm);

}

}

