namespace L11.StaticFields;

//static fields

class TimeTableReader

{

    private static string timeTableName = "timetable.txt"; //static - if nepotřebujeme hodnotu měnit a chci mít stejné pro všechny instance, nebo si toto můžu nadefinovat v static class Constants

    public string ReadTimeTable()

    {

        var fileContent = File.ReadAllText(timeTableName);

        //var fileContent = File.ReadAllText(Constants.TimeTableFileName); //if mám definováno v static class Constants

        //process the file and then

        return fileContent;

    }

}

class NonStaticTimeTableReader

{

    private static string timeTableName = "timetable.txt"; //static - if nepotřebujeme hodnotu měnit a chci mít stejné pro všechny instance

    public void SetFileName(string newFileName)

    {

        timeTableName = newFileName;  //compilátor to nebere jako chybu - umožňuje static změnit, ale změní se pro celou třídu, tomu se chceme vyhnout (není best practice)

    }

    public string GetFileName() //potřebuji metodu protože timeTableName je private

    {

        return timeTableName;

    }

    public string ReadTimeTable()

    {

        var fileContent = File.ReadAllText(timeTableName);

        //process the file and then

        return fileContent;

    }

}

class NonStaticTimeTableReaderCorrected

{

    private string timeTableName = "timetable.txt"; //nechci static - if potřebujeme hodnotu měnit

    public void SetFileName(string newFileName)

    {

        timeTableName = newFileName;  //compilátor to nebere jako chybu - umožňuje static změnit, ale změní se pro celou třídu

    }

    public string GetFileName() //potřebuji metodu protože timeTableName je private

    {

        return timeTableName;

    }

    public string ReadTimeTable()

    {

        var fileContent = File.ReadAllText(timeTableName);

        //process the file and then

        return fileContent;

    }

}

//static methods - pracuje buď se zadanými parametry (které jí předám) nebo se statickými prvky třídy (ne s objekty třídy)

class Calculator

{

    public static int Add(int a, int b) //static protože nepracuje s objektem (nepracuje s prvky objektu) ale pouze s třídou

    {

        return a + b;

    }

}

//static classes - if all fields and properties and methods are static - např. utility, config a helper classes

static class Constants

{

    public static string TimeTableFileName = "file";

}

static class CustomMath

{

    public static double Pi = 3.14; //static & readonly => const

    public static double ConvertKilometersToMiles(double km)

    {

        return km / 1.6;

    }

}

class Program

{

    static void Main(string[] args)

    {

        //var reader = new TimeTableReader();

        //var timetable = reader.ReadTimeTable();

        var reader1 = new NonStaticTimeTableReader();

        var reader2 = new NonStaticTimeTableReader();

        reader1.SetFileName("newfilename.txt"); //změníme to nejen pro reader1 ale pro všechny tj. pro celou třídu, tomu se chceme vyhnout (není best practice)

        System.Console.WriteLine(reader1.GetFileName()); //vypíše newfilename.txt

        System.Console.WriteLine(reader2.GetFileName()); //vypíše newfilename.txt

        var reader3 = new NonStaticTimeTableReaderCorrected();

        var reader4 = new NonStaticTimeTableReaderCorrected();

        reader3.SetFileName("newfilename.txt"); //změníme to jen pro reader1

        System.Console.WriteLine(reader3.GetFileName()); //vypíše newfilename.txt

        System.Console.WriteLine(reader4.GetFileName()); //vypíše timetable.txt

        System.Console.WriteLine(Constants.TimeTableFileName); //vypíše file

        System.Console.WriteLine(CustomMath.Pi); //vypíše 3,14

    }

}