C# Intermediate

Class Coupling

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Intermediate\_16\_ClassCoupling

{

    //=========

    // NOTES

    //=========

    //\*\*\*\*\*\*\*\*\*\*

    // WHAT?

    //\*\*\*\*\*\*\*\*\*\*

    // Class Coupling is a measure of how interconnected classes and subsystems are.

    // \*\*\* IMPORTANT NOTE \*\*\*

    // An Application is composed of many classes which are packaged inside various Namespaces and Assemblies.

    //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

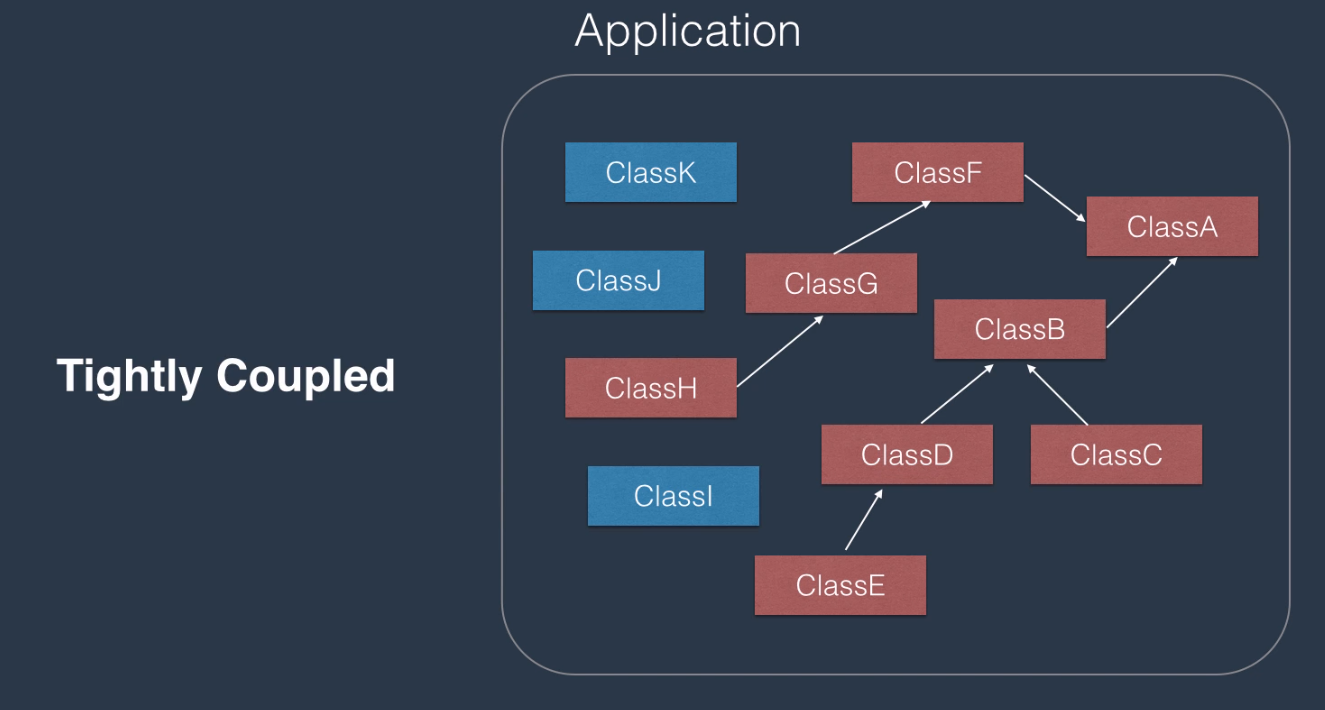
    // Tightly Coupled Design

    //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    // This is when an application's classes are tightly related or dependent on each other.

    // Tigtly coupled applications are not good because there is always a chance that changing something in one class

    // will have a negative impact on any related classes.



    //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    // Loosely Coupled Design

    //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    // When you change the code in one class, the change is isolated and does not affect any related classes.

    //\*\*\*\*\*\*\*\*\*\*

    // HOW?

    //\*\*\*\*\*\*\*\*\*\*

    // How do you design a loosely coupled Application?

    // You need to understand:

    // ENCAPSULATION

    // THE RELATIONSHIPS BETWEEN CLASSES

    // INTERFACES

    //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    // TYPES OF RELATIONSHIPS

    //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    // In general, there are 2 types of relationships between Classes:

    // INHERITANCE

    // COMPOSITION

    class Program

    {

        static void Main(string[] args)

        {

        }

    }

}